

**RANCANG BANGUN PROTOTIPE APLIKASI KONSULTASI
MENU MAKANAN BERBASIS MOBILE BAGI PENDERITA
PENYAKIT DIABETES MENGGUNAKAN METODE *HARRIS
BENEDICT***

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:

Gede Bayu Eka Bhuana

145150201111176



PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2018

PENGESAHAN

RANCANG BANGUN PROTOTIPE APLIKASI KONSULTASI MENU MAKANAN
BERBASIS MOBILE BAGI PENDERITA PENYAKIT DIABETES MENGGUNAKAN
METODE HARRIS BENEDICT

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :

Gede Bayu Eka Bhuna

NIM: 145150201111176

Skripsi ini telah diuji dan dinyatakan lulus pada
26 Juli 2018

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II

Agi Putra Kharisma, S.T, M.T

NIK: 2013048604301001

Lutfi Fanani, S.kom., M. T., M.Sc.

NIK: 2016078902171001

Mengetahui

Ketua Jurusan Teknik Informatika



Tri Astoto Kurniawan, S.T, M.T, Ph.D

NIP: 197105182003121001

PERNYATAAN ORISINALITAS

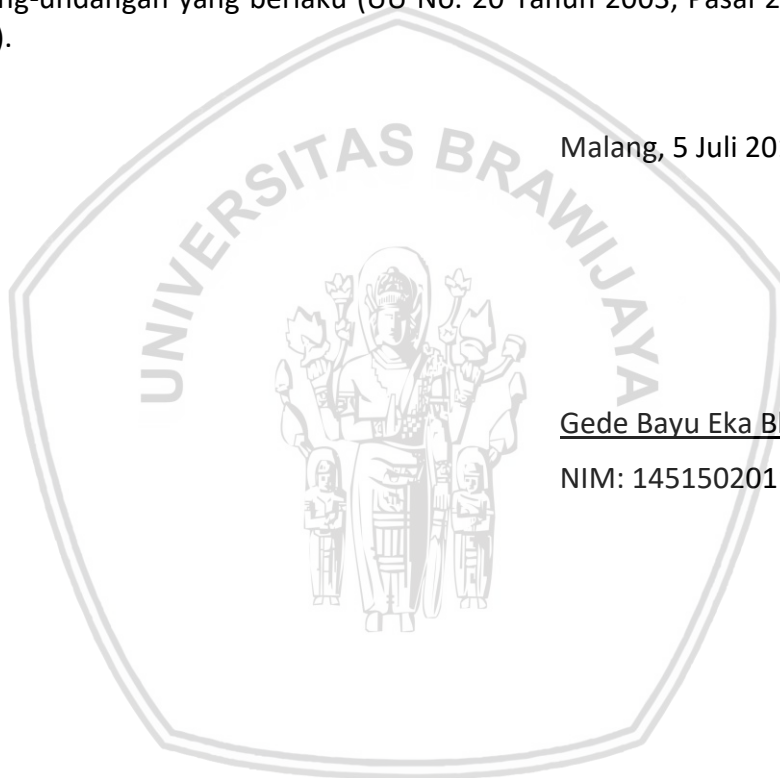
Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 5 Juli 2018

Gede Bayu Eka Bhuana

NIM: 145150201111176



KATA PENGANTAR

Puja dan Puji Syukur saya panjatkan kepada Tuhan Yang Maha Esa sehingga penulis dapat menyelesaikan skripsi yang berjudul “RANCANG BANGUN PROTOTIPE APLIKASI KONSULTASI MENU MAKANAN BERBASIS *MOBILE* BAGI PENDERITA PENYAKIT DIABETES MENGGUNAKAN METODE *HARRIS BENEDICT*”.

Untuk kesempatan ini penulis juga menyampaikan rasa terima kasih kepada pihak – pihak yang telah membantu penulis selama penyusunan skripsi, diantaranya:

1. Ida Sang Hyang Widhi Wasa yang telah memberi kemudahan dalam semua proses penulisan skripsi ini.
2. Kepada kedua orang tua penulis, yaitu Komang Ariyasa dan Ibu Yeni Lestari beserta keluarga besar yang selalu memberikan segala masukan, do’a, motivasi dan semangat yang tidak terputus.
3. Bapak Agi Putra Kharisma, S.T, M.T selaku dosen pembimbing I yang telah meluangkan waktu untuk memberikan masukan, ilmu, serta saran yang bermanfaat dalam proses penyelesaian skripsi ini.
4. Bapak Lutfi Fanani, S.kom., M. T., M.sc. selaku dosen pembimbing II yang juga telah meluangkan waktu untuk memberikan masukan, ilmu, serta saran yang bermanfaat dalam proses penyelesaian skripsi ini.
5. Bapak Wayan Firdaus Mahmudy, S.Si, M.T, Ph.D, Bapak Ir. Heru Nurwasito, M.Kom, Bapak Drs. Mardji, M.T, dan Bapak Edy Santoso, S.Si, M.Kom selaku Dekan, Wakil Dekan 1, Wakil Dekan 2, dan Wakil Dekan 3 Fakultas Ilmu Komputer, Universitas Brawijaya.
6. Bapak Tri Astoto Kurniawan, S.T, M.T, Ph.D dan Bapak Agus Wahyu Widowo, S.T, M.Cs selaku Ketua Jurusan Teknik Informatika dan Kepala Program Studi Teknik Informatika Fakultas Ilmu Komputer, Universitas Brawijaya.
7. Seluruh Dosen Fakultas Ilmu Komputer, Universitas Brawijaya atas kesediaannya dalam mengajarkan dan membagikan ilmu yang bermanfaat bagi penulis.
8. Ibu Maria Ika Rosita sebagai *Expert* dibidang gizi masyarakat yang senantiasa membantu dan menuntun penulisan skripsi ini.
9. Semua pihak yang tidak semuanya bisa dituliskan disini yang terlibat secara langsung maupun tidak langsung dalam proses pengerjaan skripsi maupun sebagai pemberi semangat dan motivasi.
10. Teman – teman bestman yang selalu mensupport dalam pembuatan penelitian kali ini.
- 11.

Penulis menyadari masih banyak kekurangan dalam penyusunan skripsi ini baik dalam teknik penyajian materi maupun pembahasan. Demi kesempurnaan penelitian skripsi ini, saran dan kritik yang sifatnya membangun sangat penulis harapkan. Semoga karya tulis ini bermanfaat dan dapat memberikan sumbangan yang berarti bagi pihak yang membutuhkan.

Malang, 05 Juli 2018

Penulis

Gede Bayu Eka Bhuana



ABSTRAK

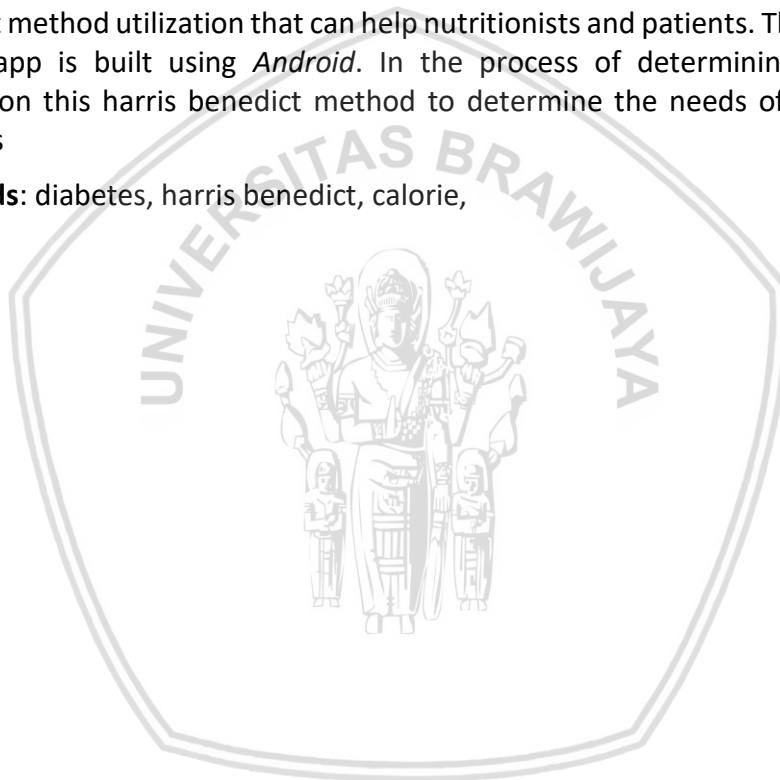
Sample Registration Survei tahun, 2014 mengatakan diabetes menjadi penyakit pembunuh nomor tiga di Indonesia. Timbulnya Komplikasi yang disebabkan oleh penyakit diabetes menjadipenyebab kematian tertinggi ketiga di Negara kita Indonesia (Depkes, 2008). Dimana menurut Prof Sidartawan, Resiko diabaets meningkat 30 persen setiap tahunnya sehingga dianjurkan bagi kita untuk melakukan pemeriksaan gula darah setahun sekali. Akan Tetapi Mahalnya membayar seorang ahli konsultan makanan bagi penderita penyakit diabetes telah menjadi salah satu kesulitan kita untuk mengatur pola makan kita sendiri. Harga jasa ahli konstultan makanan mencapai satu juta lebih sekali pertemuan. Oleh faktor – faktor tersebut dan juga wawancara yang telah saya lakukan dengan lulusan - lulusan baru ahli gizi ub tentang kekurangannya mereka akan pasien penulis termotivasi untuk membuat sebuah prototipe aplikasi berbasis mobile dengan pemanfaatan metode *harris benedict* yang dapat membantu ahli gizi dan pasien. Prototipe aplikasi ini dibangun berbasis *mobile android*. Dalam proses penentuan kalori aplikasi ini memanfaatkan metode *harris benedict* untuk menentukan kebutuhan kalori bagi penderita diabetes. Nantinya akan dihasilkan sebuah aplikasi yang dapat membantu pasien penderita penyakit diabetes untuk menemukan ahli gizi dan juga membantu ahli gizi itu sendiri untuk mendapatkan pasien – pasein yang ingin melakukan konsultasi menu makanan.

Kata kunci: Diabetes, Harris Benedict, Kalori

ABSTRACT

Sample Registration Surve year, 2014 says diabetes becomes the third killer disease in Indonesia. The incidence of complications caused by diabetes became the third highest cause of death in our country Indonesia (MOH, 2008). Where according to Prof. Sidartawan, Risks diabaets increase 30 percent every year so it is recommended for us to do blood sugar check once a year. But it is expensive to pay a food consultant for diabetics has become one of our difficulties to regulate our own diet. The price of food constituent *expert* services reaches one million more meetings. By these factors andalso the interviews I have done with recent graduates of nutritionist *experts* about their shortcomings will be patient writers motivated to create a prototype of mobile-based applications with harris benedict method utilization that can help nutritionists and patients. The prototype of this app is built using *Android*. In the process of determining its calorie application this harris benedict method to determine the needs of calories for diabetics

Keywords: diabetes, harris benedict, calorie,



BAB 1 PENDAHULUAN

1.1 Latar Belakang

Banyak penyakit yang harus kita waspadai, salah satunya adalah penyakit diabetes. Diabetes mellitus adalah penyakit yang diakibatkan oleh kelainan sekresi *insulin*. Penyakit ini sangat sulit dideteksi dan dikatakan onset atau terjadi 7 tahun sebelum diagnosis itu diberitahu.

Yang mengakibatkan sulitnya penanganan secara tepat waktu (Soegondo, et al., 2005). Biasanya banyak pengidap diabetes militus baru mengetahui dia mengidap penyakit diabetes setelah terjadi komplikasi penyakit yang disebabkan penyakit diabetes itu sendiri, hal itu diungkapkan oleh Prof.DR.Dr. Sidartawan Soegondo, SpPD, KEMD, FACE di kantornya di Bagian Metabolik dan Endokrin, FKUI/RSCM.

Sample Registration Surve tahun, 2014 mengatakan diabetes menjadi penyakit pembunuh nomor tiga di Indonesia. Timbulnya Komplikasi yang disebabkan oleh penyakit diabetes menjadipenyebab kematian tertinggi ketiga di Negara kita Indonesia (Depkes, 2008). Dimana menurut Prof Sidartawan, Resiko diabaets meningkat 30 persen setiap tahunnya sehingga dianjurkan bagi kita untuk melakukan pemeriksaan gula darah setahun sekali.

Penyakit diabetes tidak bisa disembuhkan sepenuhnya, oleh karena itu sangat wajib bagi kita untuk memulai pola hidup sehat. Salah satu dari banyak solusi untuk mengontrol penyakit diabetes ini agar tidak terjadi komplikasi penyakit lainnya adalah mengatur pola makan anda sendiri (Joel Fuhrman, et al., 2003) Makanan yang di konsumsi oleh penderita diabetes harus mampu menjaga kadar gula darah agar tidak berlebihan.

Mahalnya membayar seorang ahli gizi sebagai konsultan menu makanan bagi penderita penyakit diabetes juga menjadi salah satu kesulitan kita untuk mengatur pola makan kita sendiri. Harga jasa ahli konstultan makanan mencapai satu juta lebih sekali pertemuan. Diluar negeri sudah banyak apalikasi berbayar yang dapat membantu anda untuk menentukan menu makanan yang cocok dengan keadaan kesahatan anda sendiri. Namun kini di indonesia sangat sulit untuk menemukan aplikasi yang dapat mengatur menu makanan yang sesuai bagi penyandang diabetes militus. Menurut WHO, 2010 Banyak penderita penyakit diabetes memerlukan suatu media yang dapat membantu mereka untuk mengatur menu makanan, selain konsultasi ke ahli gizi yang menguras biaya yang cukup banyak.

Oleh faktor – faktor tersebut dan juga wawancara yang telah saya lakukan dengan lulusan - lulusan baru ahli gizi ub tentang kekurangannya mereka akan pasien. Penulis dalam penelitian ini termotivasi untuk membuat sebuah prototipe aplikasi *berbasis* mobile yang dapat membantu ahli gizi dan pasien. Sehingga para penderita penyakit diabetes tidak perlu menguras kantong cukup dalam untuk konsultasi ke ahli gizi untuk menentukan menu makanan yang tepat bagi dirinya.

Diduga aplikasi ini dapat menurunkan persentase timbulnya komplikasi - komplikasi penyakit lain bagi penyandang diabetes karena pola makan yang kurang baik.

1.2 Identifikasi Masalah

Beberapa masalah yang dapat saya identifikasi dalam penelitian ini adalah sebagai berikut:

1. Para penderita diabetes susah untuk menentukan menu makanan yang sesuai untuk penyakit yang ia derita
2. Mahalnya biaya ahli gizi bagi penderita diabetes untuk melakukan konsultasi
3. Kurangnya waktu yang dimiliki penderita penyakit diabetes untuk melakukan konsultasi kepada ahli gizi
4. Susahnya lulusan - lulusan baru ahli gizi ub untuk mendapatkan pasien dalam melakukan konsultasi dengan mereka

1.3 Rumusan Masalah

Rumusan masalah yang dapat saya temukan dalam penelitian ini ialah sebagai berikut:

1. Bagaimana cara menentukan kalori yang dibutuhkan khusus bagi penderita penyakit diabetes?
2. Bagaimana cara menentukan menu makanan yang tepat bagi penderita penyakit diabetes?
3. Bagaimana rancangan dan implementasi aplikasi perangkat bergerak yang dapat membantu lulusan - lulusan ahli gizi ub baru untuk mendapatkan pasien?
4. Bagaimana rancangan dan implementasi aplikasi perangkat bergerak yang dapat membantu pasien penderita penyakit diabetes untuk mencari ahli gizi sebagai konsultan menu makanan?

1.4 Tujuan Penelitian

Tujuan penelitian merupakan jawaban atau sasaran yang ingin dicapai penulis dalam sebuah penelitian. Oleh sebab itu, tujuan umum penelitian ini adalah: membuat aplikasi konsultasi penentu gizi makanan.

Sedangkan tujuan khusus yang dapat diperoleh dari penelitian ini adalah:

1. Membantu para penderita penyakit diabetes untuk mengetahui kalori yang mereka butuhkan
2. Menyediakan fasilitas bagi penderita penyakit diabetes untuk menentukan menu makanan yang tepat
3. Membantu lulusan - lulusan baru ahli gizi Universitas Brawijaya untuk mendapatkan pasien.
4. Membantu pasien penderita penyakit diabetes untuk menemukan ahli gizi sebagai konsultan menu makanan

1.5 Manfaat Penelitian

Beberapa manfaat yang diharapkan oleh peneliti adalah manfaat teoritis dan manfaat praktis. Berikut adalah penjelasan dari manfaat tersebut:

a. Manfaat Teoritis

Diharapkan penelitian ini mampu untuk memperkaya teori teori yang sudah ada kemudian mengembangkan lagi teori teori yang sedang berkembang sekarang

b. Manfaat Praktis

1. Bagi Penulis

Penulis mampu mengimplementasi ilmu yang mereka dapat dalam kasus nyata. Sehingga mahasiswa dapat membandingkan seberapa besar hubungan teori yang mereka pelajari dengan kenyataan yang ada di lapangan kerja

2. Bagi Penderita Penyakit diabetes

Masyarakat penderita penyakit diabetes dipermudah dalam menentukan menu makanan yang tepat dengan adanya aplikasi ini.

3. Bagi Lulusan Baru Ahli Gizi UB

Para lulusan Baru Ahli Gizi UB mendapatkan sebuah media yang dapat membantu mereka untuk menemukan pasien - pasien baru untuk melakukan konsultasi dengan mereka

1.6 Batasan Penelitian

Aplikasi konsultasi menu makanan berbasis web bagi penderita penyakit diabetes memiliki cakupan yang sangat luas, sementara waktu diberikan kepada penulis untuk melakukan penelitian sangat terbatas. Karena itu penulis merasa perlu untuk melakukan pembatasan penelitian dalam hal-hal sebagai berikut:

- Penelitian ini hanya menyediakan menu makanan yang disediakan hanya ditujukan bagi penderita penyakit diabetes.
- Penelitian ini hanya menggunakan metode *Harris Benedict* sebagai acuan untuk menentukan menu makanan tiap harinya bagi penderita penyakit diabetes.
- Aplikasi yang dirancang didalam penelitian ini hanya berjalan di *platform android*.

1.7 Sistematika pembahasan

Berikut merupakan sistem matika untuk melanjutkan penelitian pada tahap berikutnya:

BAB I PENDAHULUAN

Bab ini berisikan uraian dari latar belakang, rumusan masalah,

tujuan dan sasaran studi, ruang lingkup studi, sistematika penulisan dan kerangka berfikir.

BAB II TINJAUAN PUSTAKA

Bab ini berisikan kajian terhadap beberapa teori dan referensi yang menjadi landasan dalam mendukung studi penelitian ini, diantaranya adalah teori mengenai Penyakit diabetes, teori penghitungan kalori makanan bagi penderita penyakit diabetes

BAB III METODOLOGI PENELITIAN

Bab ini berisikan mengenai metode penelitian, metode pendekatan, dan metode pengumpulan data yang dilakukan dalam penelitian ini

BAB IV ANALISA KEBUTUHAN DAN PERANCANGAN

Didalam bab ini menjelaskan mengenai kebutuhan – kebutuhan apa saja yang dimiliki dalam tahap pembuatan aplikasi ini serta bagaimana perancangannya di depannya nanti.

BAB III IMPLEMENTASI DAN PEMBAHASAN

Bab ini berisikan mengenai implementasi dari perancangan yang telah dibuat sebelumnya berserta pembahasan dari implementasi tersebut.

BAB III PENGUJIAN DAN ANALISIS

Bab ini berisikan mengenai proses pengujian dan analisis dari pengujian tersebut apakah aplikasi yang dikembangkan dapat dikatakan berhasil atau tidak.

BAB III PENUTUP

Bab ini berisikan mengenai kesimpulan dari penelitian serta saran untuk pengembangan aplikasi selanjutnya.

BAB 2 TINJAUAN PUSTAKA

2.1 Diabetes

2.1.1 Pengertian Diabetes

Secara garis besar diabetes merupakan penyakit yang disebabkan karena seseorang terindikasi memiliki kadar gula yang tinggi didalam darah, hal tersebut disebabkan karena sekresi *insulin* didalam tubuh mengalami gangguan. Terdapatnya kadar gula yang tinggi dalam air kencing kita bisa menjadi tanda-tanda gejala awal penyakit Diabetes Melitus (Depkes, 2008).

Insulin merupakan salah satu hormon yang diproduksi oleh pankreas didalam tubuh, hormon *insulin* memiliki fungsi untuk mengendalikan kadar gula dalam darah menjaga kadar gula darah agar tidak terlalu rendah maupun tinggi. Turunnya fungsi sekresi *insulin* biasanya disebabkan oleh kerusakan sel beta pankreas (Merck, 2008).

Tubuh Penderita penyakit diabetes mellitus tidak dapat memproduksi ataupun tidak merespon hormon *insulin* yang dihasilkan oleh organ pancreas tersebut. Kekurangan *insulin* menyebabkan tubuh tidak mampu mengolah glukosa menjadi sumber energi bagi sel untuk melakukan tugasnya.

Dan pada akhirnya respon yang diterima oleh tubuh adalah rasa lapar dan haus. Karena rasa lapar dan haus ini penderita penyakit diabetes cenderung mengonsumsi semakin banyak karbohidrat, maka akan semakin tinggi penumpukan glukosa dalam darah. Kondisi inilah yang kemudian disebut sebagai penyakit kencing manis atau Diabetes Melitus

2.1.2 Gejala Diabetes

Sebelum seseorang dinyatakan mengidap penyakit diabetes terdapat beberapa gejala diabetes yang patut diwaspadai, Berikut adalah sedikit penjelasan mengenai Beberapa gejala penyakit diabetes yang ada yaitu:

1. Berat badan berkurang

Salah satu gejala diabetes adalah berat badan yang berkurang secara drastis. Banyak orang menginginkan berat badan mereka turun. Tapi hal ini menjadi berbeda apa bila berat badan anda turun dengan cepat. Bisa saja faktor pemicu turunnya berat badan yang cepat adalah karena diabetes (Depkes, 2008).

2. Lebih sering buang air kecil

Gejala lain yang perlu diwaspadai adalah lebih seringnya anda untuk buang air kecil. Hal ini merupakan salah satu gejala yang sering terjadi pada penderita penyakit diabetes (Depkes, 2008).

3. Rasa lapar yang meningkat

Para penderita penyakit diabetes sering mengalami rasa lapar dan haus yang tidak bias dikendalikan, hal ini disebabkan karena penyerapan nutrisi

karbohidrat yang tidak sempurna. Menyebabkan sel terus menerus memberi sinyal untuk meminta asupan karbohidrat yang lebih daripada biasanya (Depkes, 2008).

2.1.3 Cara Mengontrol Penyakit Diabetes

Banyak cara yang dapat kita tempuh untuk Mencegah Penyakit kencing manis. Salah satu caranya yaitu dengan mengatur pola makan kita sendiri. Kita dapat menghindari jenis – jenis makanan mengandung banyak glukosa, dan berbagai jenis olahan makanan yang tinggi glukosa seperti manisan, nasi dan hal hal manis lainnya. Penting juga untuk rajin berolah raga dan melakukan aktivitas fisik minimal 3 kali seminggu (Herdaningsih, 2006).

Terutama bagi anda yang memiliki gejala penyakit diabetes tersebut, terlebih apabila terdapat keluarga anda yang menderita Kencing manis berarti resiko untuk terkena diabetes dua kali lebih besar dibandingkan mereka yang tidak memiliki histori keluarga yang memiliki penyakit diabetes. Karena penyakit diabetes ini terkenal dengan penyakit bawaan atau warisan. Orang tua atau saudara yang memiliki histori penyakit diabetes bisa menjadi peringatan bagi kita untuk menjaga pola makanan dan pola olahraga yang teratur. Karena bisa saja penyakit diabetes itu diwariskan kepada kita.

Selain itu upaya untuk mengontrol berat badan juga penting dalam rangka mencegah terjadinya penyakit diabetes ini, sangat disarankan untuk rutin melakukan pengecekan kadar gula darah anda sendiri (Merk, 2008). Hal tersebut dilakukan untuk mengetahui berapa tinggi glukosa dalam darah dan upaya sedini mungkin untuk bisa melakukan pencegahan. Karena bagaimanapun mencegah lebih baik daripada mengobati.

2.2 Gizi

Pengertian gizi menurut Supriasa adalah suatu proses organisme menggunakan makanan yang dikonsumsi secara normal melalui proses pencernaan, absorpsi, transportasi, penyimpanan, metabolisme dan pengeluaran zat-zat yang tidak digunakan untuk mempertahankan kehidupan, pertumbuhan dan fungsi normal dari organ-organ, serta menghasilkan energi. (Supriasa, dkk, 2002).

Dimana sebuah Status gizi merupakan keadaan yang diakibatkan oleh status keseimbangan antara jumlah asupan (intake) zat gizi dan jumlah yang dibutuhkan (requirement) oleh tubuh untuk berbagai fungsi biologis: (pertumbuhan fisik, perkembangan, aktivitas, pemeliharaan kesehatan, dan lainnya). (Suyatno, 2009). Status gizi adalah ekspresi dari keadaan keseimbangan dalam bentuk variabel tertentu atau perwujudan dari nutrisi dalam bentuk variabel tertentu (Supriasa, dkk, 2001). Pada gilirannya, zat gizi tersebut menyediakan tenaga bagi tubuh, mengatur proses dalam tubuh dan membuat lancarnya pertumbuhan serta memperbaiki jaringan tubuh. Beberapa zat gizi yang disediakan oleh pangan tersebut disebut zat gizi essential, mengingat kenyataan bahwa unsur-unsur tersebut tidak dapat dibentuk dalam tubuh, setidaknya-tidaknya

dalam jumlah yang diperlukan untuk pertumbuhan dan kesehatan yang normal. Jadi zat gizi esensial yang disediakan untuk tubuh yang dihasilkan dalam pangan, umumnya adalah zat gizi yang tidak dibentuk dalam tubuh dan harus disediakan dari unsur-unsur pangan di antaranya adalah asam amino esensial. Semua zat gizi esensial diperlukan untuk memperoleh dan memelihara pertumbuhan, perkembangan dan kesehatan yang baik.

Tubuh manusia memerlukan sejumlah pangan dan gizi secara tetap, sesuai dengan standar kecukupan gizi, namun kebutuhan tersebut tidak selalu dapat terpenuhi. Penduduk yang miskin tidak mendapatkan pangan dan gizi dalam jumlah yang cukup. Mereka menderita lapar pangan dan gizi, mereka menderita gizi kurang. (Sri Handayani, 1996). Keadaan gizi seseorang merupakan gambaran apa yang dikonsumsi dalam jangka waktu yang cukup lama. Bila kekurangan itu ringan, tidak akan dijumpai penyakit defisiensi yang nyata, tetapi akan timbul konsekuensi fungsional yang lebih ringan dan kadang-kadang tidak disadari kalau hal tersebut karena faktor gizi. (Ari Agung, 2002).

2.3 Kalori

2.3.1 Pengertian Kalori

Pengertian Kalori merupakan salah satu kandungan dalam makanan yang bermanfaat bagi tubuh sebagai asupan energy. Setiap kebutuhan kalori seseorang berbeda menurut usia, tinggi badan, berat badan, jenis kelamin, dan tingkat aktivitas perhari. Jika seseorang mengonsumsi makanan yang mengandung kalori berlebih, maka hal tersebut dapat mengakibatkan kegemukan (Deddy Pamudji:2011).

Menurut Poerwadarminta (1976:438) bahwa kalori adalah Satuan Ukuran Panas (yaitu: panas yang diperlukan untuk memanaskan 1 kg air hingga suhunya naik 10 C) Satuan energi dinyatakan dalam unit panas atau kkalori (kkal). Satu kkalori adalah jumlah panas yang diperlukan untuk menaikkan suhu 1 kg air sebanyak 10 C. Pada umumnya lebih diistilahkan kalori. Satu kalori adalah 0,001 kkal. Istilah kkalori digunakan untuk menyatakan suatu kkalori dalam jumlah tertentu, sedangkan pengertian kalori lebih digunakan untuk menyatakan suatu energy. Sedangkan kilorimetri adalah ukuran jumlah panas yang dihasilkan.

Nilai energi sebuah makanan dan energy yang dikeluarkan dalam sehari oleh seseorang diukur menggunakan metode kalorimetri dan diucapkan dalam kkalori. Pengukuran secara langsung pada jumlah panas yang dihasilkan, disebut kalorimetri langsung, dan pengukuran secara tak langsung pada jumlah panas yang dihasilkan disebut kalorimetri tidak langsung (Almatsier, 2009:133).

2.3.2 Kebutuhan Kalori

Kalori yang dibutuhkan manusia berbeda – beda antara individu satu dengan individu lainnya. Ada beberapa faktor yang menentukan kebutuhan kalori manusia perharinya, beberapa diantaranya yaitu berat badan dan tinggi badan. Kebutuhan kalori seseorang menurut FAO/WHO (1985) adalah konsumsi energi

berasal dari makanan yang diperlukan untuk menutupi pengeluaran energi seseorang. Kebutuhan kalori yang anda butuhkan dapat dihitung dengan menentukan *basal metabolic rate* dan aktivitas fisik.

Kebutuhan kalori perhari setiap manusia berbeda – beda. Kemenkes RI merekomendasikan angka kecukupan gizi pada wanita dewasa yang berumur 16 – 30 tahun pada umumnya berkisar antara 2,125 sampai 2,250 kalori per hari. Sementara pria dewasa dengan rentang umur 16 – 30 tahun yaitu 2625-2725 kalori per hari.

2.4 *Harris Benedict*

Metode *Harris Benedict* merupakan salah satu dari banyak metode yang digunakan untuk menentukan berapa banyak kalori perhari yang kita butuhkan untuk memberikan energi pada tiap kegiatan kita setiap hari (Almatseir, 2008).

Dari perhitungan *harris benedict* tersebut akan dihasilkan BMR atau *Basal Metabolic Rate* dan kebutuhan manusia dalam kilo kalori. Butuh beberapa variable tertentu untuk melakukan perhitungan kalori manusia perharinya menggunakan metode *harris benedict*, beberapa contohnya yaitu berat badan, tinggi badan, umur dan faktor aktivitas.

2.4.1 Faktor Aktivitas

Menurut Sunita Almatsier (Almatsier, 2003) faktor aktivitas merupakan faktor – faktor yang mempengaruhi kebutuhan kalori manusia dari aktivitas apa saja yang mereka lakukan. Terdapat pembagian kategori dari faktor aktivitas itu sendiri yaitu ringan, sedang dan berat. Tabel 2.1 menjelaskan lebih detail bagaimana pembagian faktor aktivitas lebih rincinya.

Tabel 2.1 Tabel Faktor Aktivitas

Sumber (Almatsier, et al (2003))

| No | Aktifitas | Faktor | No | Jenis trauma / stres | Faktor |
|----|-------------------------------|--------|----|--|--------|
| 1. | Istirahat di tempat tidur | 1,2 | 1. | Tidak ada stres, pasien dalam keadaan gizi baik | 1,3 |
| 2. | Tidak terikat di tempat tidur | 1,3 | 2. | Stres ringan: peradangan saluran cerna, kanker, bedah elektif, trauma kerangka moderat | 1,4 |
| | | | 3. | Stres sedang: sepsis, bedah tulang, luka bakar, trauma kerangka mayor | 1,5 |

| | | | | | |
|--|--|--|----|--|-----|
| | | | 4. | Stres berat: trauma multiple, sepsis, dan bedah multisistem | 1,6 |
| | | | 5. | Stres sangat berat: luka kepala berat, sindroma penyakit pernapasan akut, luka bakar dan sepsis Luka bakar sangat berat | 1,7 |

2.4.2 Perhitungan *Harris benedict*

Kebutuhan kalori manusia perharinya akan berbeda antara individu satu dan yang lainnya. Salah satu cara untuk menghitung kebutuhan kalori manusia tersebut ialah menggunakan rumus *harris benedict*. Rincian dari penggunaan rumus *Harris benedict* adalah sebagai berikut

(*BMR*) Perempuan : $655 + (9,6 \times BB) + (1,8 \times TB) - (4,7 \times U)$

(*BMR*) Laki-laki : $66 + (13,7 \times BB) + (5 \times TB) - (6,8 \times U)$

Keterangan:

BB : Berat badan dalam Kg

TB : Tinggi badan dalam Cm

U : Usia dalam Tahun

BMR : *Basal metabolic rate*

Dari nilai *BMR* tersebut kita dapat mengetahui kebutuhan kalori seseorang dengan mengkalkulasi *BMR* tersebut dengan nilai faktor aktivitas seseorang. Kemudian didapatkanlah kebutuhan kalori individu tersebut perhari dalam kilo kalori.

2.5 Kalori Pada Makanan

Masing – masing makanan memiliki kandungan kalori tersendiri. Besarnya kalori makanan tergantung pada berat atau takar saji makanan itu. Membutuhkan sebuah penelitian tersendiri untuk dapat mengetahui kalori yang terdapat pada makanan. Tabel 2.2 akan menjelaskan jumlah kalori yang terkandung didalam beberapa menurut hasil riset dari departemen kesehatan tahun 2011

Tabel 2.2 Tabel Jumlah Kalori Pada Makanan

Sumber (Riset Departemen Kesehatan 2011)

| DAFTAR TABEL KALORI & UNIT | | | |
|----------------------------|------------|--------|------|
| Makanan Pokok Golongan A | | | |
| Nama Masakan | Berat (gr) | Kalori | Unit |

| | | | |
|-------------------------|-----|------|------|
| Jagung Rebus | 250 | 90,2 | 1 |
| Kentang Rebus | 200 | 166 | 2 |
| Ketan Putih | 120 | 217 | 2,75 |
| Ketupat | 160 | 32 | 0,5 |
| Lontong | 200 | 38 | 0,5 |
| Nasi Putih | 100 | 175 | 2,25 |
| Nasi Putih Kentucky | 225 | 349 | 4,25 |
| Roti Tawar Serat Tinggi | 60 | 149 | 1,75 |
| Singkong Rebus | 100 | 146 | 1,75 |
| Talas Rebus | 100 | 98 | 1,25 |
| Ubi Rebus | 100 | 125 | 1,5 |

Makanan Pokok Golongan B

| | | | |
|-------------|-----|-----|------|
| Bubur | 200 | 44 | 0,5 |
| Crakers | 50 | 229 | 2,75 |
| Makaroni | 25 | 91 | 1,25 |
| Mie Instant | 50 | 168 | 2 |
| Nasi Tim | 100 | 88 | 1 |
| Nasi Uduk | 200 | 506 | 6,25 |
| Roti Tawar | 50 | 128 | 1,5 |

Makanan Pokok Golongan C

| | | | |
|----------------|-----|-----|------|
| Bihun Goreng | 150 | 296 | 3,75 |
| Bubur Ayam | 200 | 165 | 2 |
| Bubur Sum-Sum | 100 | 178 | 2,25 |
| Kentang Goreng | 150 | 211 | 2,75 |
| Mie Goreng | 200 | 321 | 4 |
| Nasi Goreng | 100 | 267 | 3,25 |
| Soun Goreng | 100 | 263 | 3,25 |
| Spaghetti | 300 | 642 | 8 |
| Tape singkong | 150 | 260 | 3,25 |

| Sayuran Golongan A | | | |
|--------------------|-----------|--------|------|
| Nama Masakan | Berat(gr) | Kalori | Unit |
| Acar Kuning | 75 | 53 | 0,5 |
| Bening Bayam | 50 | 18 | 0,25 |
| Cah Labu Siam | 100 | 41,6 | 0,5 |
| Sayur Asam | 100 | 88 | 1 |
| Sop Ayam Kombinasi | 100 | 95 | 1,25 |
| Sop Bayam | 50 | 78 | 1 |
| Sop Kimlo | 100 | 104 | 1,25 |
| Sop Mutiara Jagung | 100 | 113 | 1,5 |
| Asop Oyong Misoa | 100 | 106 | 1,25 |

| Sayuran Golongan B | | | |
|----------------------------|-----|-----|------|
| Sayur Lodeh | 100 | 61 | 0,75 |
| Cah Jagung Putren | 100 | 59 | 0,75 |
| Cah Jkacang Panjang | 100 | 72 | 1 |
| Sop Oyong Telur Puyuh | 100 | 134 | 1,75 |
| Setup Kentang Buncis | 100 | 95 | 1 |
| Tumis Buncis | 100 | 52 | 1,5 |
| Tumis Daun Singkong | 120 | 151 | 1,75 |
| Tumis Kc. Panjang + Jagung | 125 | 118 | 1,75 |

| Lauk Pauk Golongan C | | | |
|---------------------------|----|-------|------|
| Abon Sapi | 50 | 158 | 2 |
| Ayam Goreng Kecap | 75 | 358,8 | 4,5 |
| Ayam Panggang | 80 | 385,6 | 4,75 |
| Chicken Wing / Sayap Ayam | 50 | 63,6 | 0,75 |
| Daging Balado | 50 | 147 | 1,75 |

| | | | |
|----------------------------------|-----|-------|------|
| Dendeng Balado | 40 | 338 | 4,25 |
| Gulai Ayam | 100 | 165,3 | 2 |
| Gulai Cumi | 100 | 183 | 2,25 |
| Gulai Kepala Ikan Kakap | 320 | 218,8 | 2,75 |
| Gulai Limpa | 60 | 294 | 3,5 |
| Gulai Tunjang | 80 | 251 | 3 |
| Ikan Kembung Balado | 125 | 236,7 | 3 |
| Ikan Teri | 50 | 213 | 2,75 |
| Kakap Goreng Tepung | 80 | 119 | 1,5 |
| Kakap Panir | 75 | 220 | 2,75 |
| Keripik Tempe | 25 | 68 | 0,75 |
| Meat Ball / Daging Cincang Bulat | 50 | 168 | 2 |
| Ayam Kentucky Paha Atas | 150 | 194,5 | 2,5 |
| Perkedel Jagung | 50 | 108 | 1,25 |
| Perkedel Kentang | 50 | 123 | 1,5 |
| Pu Yung Hai | 50 | 114 | 1,5 |
| Rendang Daging | 75 | 285,5 | 3,5 |
| Sate Ayam | 100 | 466 | 6 |
| Ayam Kentucky Sayap | 150 | 116 | 1,5 |
| Semur Ayam | 50 | 177,8 | 2,25 |
| Sop Sapi | 260 | 227 | 2,75 |
| Tahu Goreng | 100 | 111 | 1,5 |
| Tahu Isi | 150 | 124 | 1,5 |
| Tahu Sumedang | 100 | 113 | 1,5 |
| Telur dadar | 75 | 188 | 2,25 |

| Minuman Golongan A | | | |
|--------------------|-----------|--------|------|
| Nama Minuman | Berat(gr) | Kalori | Unit |
| Teh (Cangkir) | 150 | 80 | 1 |
| Kopi (Cangkir) | 1 | 18 | 0,25 |

| | | | |
|-------------------|-----|----|------|
| Jus Tomat (Gelas) | 100 | 20 | 0,25 |
| Jus Melon (Gelas) | 150 | 35 | 0,5 |

| Minuman Golongan B | | | |
|-------------------------|-----|-----|------|
| Es Kelapa Muda (Gelas) | 100 | 42 | 0,5 |
| Es Cendol | 100 | 168 | 2 |
| Susu Skim | 15 | 54 | 0,75 |
| Coca Cola Diet (Kaleng) | 1 | 1 | 0 |

| Masakan Siap Saji Golongan A | | | |
|------------------------------|-----------|--------|------|
| Nama Masakan | Berat(gr) | Kalori | Unit |
| Asinan | 250 | 208 | 2,5 |
| Toge Goreng | 250 | 243 | 3 |

| Masakan Siap Saji Golongan B | | | |
|------------------------------|-----|-----|------|
| Gado – Gado | 150 | 295 | 3,75 |
| Ketoprak | 250 | 153 | 2 |
| Pempek | 200 | 384 | 4,75 |
| Rawon | 160 | 331 | 4 |
| Soto Ayam | 100 | 101 | 1,25 |
| Soto Padang | 100 | 127 | 1,5 |
| Tongseng | 120 | 331 | 4 |

| Masakan Siap Saji Golongan C | | | |
|------------------------------|-----|-----|------|
| Hamburger | 125 | 257 | 3,25 |
| Kerupuk Palembang | 50 | 168 | 2 |
| Kerupuk Udang | 20 | 72 | 1 |
| Mie Bakso | 200 | 302 | 3,75 |
| Nasi Tim Ayam | 420 | 588 | 7,25 |
| Pizza | 125 | 163 | 2 |

| | | | |
|--------------|-----|-----|------|
| Sate Kambing | 180 | 729 | 9 |
| Sayur Krecek | 175 | 249 | 3 |
| Siomay | 100 | 361 | 3,75 |
| Soto Betawi | 150 | 135 | 1,75 |
| Soto Makasar | 150 | 525 | 6,5 |
| Soto Sulung | 150 | 86 | 1 |

| Buah -Buahan Golongan A | | | |
|-------------------------|-----------|--------|------|
| Nama Masakan | Berat(gr) | Kalori | Unit |
| Apel | 160 | 92 | 1 |
| Apel Merah | 140 | 82 | 1 |
| Belimbing | 160 | 80 | 0,75 |
| Duku | 200 | 81 | 1 |
| Jambu Air | 60 | 35,4 | 0,5 |
| Jambu Biji | 320 | 157 | 2 |
| Jeruk Medan | 140 | 46 | 0,5 |
| Jeruk Pontianak | 150 | 67 | 0,75 |
| Jeruk Sunkist | 200 | 40 | 0,5 |
| Mangga Manalagi | 100 | 72 | 1 |
| Nanas | 200 | 104 | 1,25 |
| Pepaya | 100 | 46 | 0,5 |
| Pir | 200 | 80 | 1 |
| Pisang Rebus | 125 | 136,5 | 1,75 |
| Salak | 150 | 63,6 | 0,75 |
| Semangka | 150 | 48 | 0,5 |

| Buah – Buahan Golongan B | | | |
|--------------------------|-----|----|------|
| Alpukat | 100 | 85 | 1 |
| Anggur | 125 | 60 | 0,75 |
| Lengkeng | 100 | 79 | 1 |

| | | | |
|--------------------|-----|------|------|
| Melon | 120 | 46 | 0,5 |
| Mangga Harum Manis | 300 | 90 | 1 |
| Pir Hijau | 200 | 105 | 1,25 |
| Pisang Ambon | 100 | 74,2 | 1 |
| Pisang Barangan | 200 | 236 | 3 |
| Pisang Mas | 125 | 120 | 1,5 |
| Pisang Raja | 150 | 126 | 1,5 |
| Sirsak | 125 | 55 | 0,25 |

| Buah – Buahan Golongan C | | | |
|--------------------------|-----|-----|------|
| Durian Montong | 100 | 134 | 1,5 |
| Rambutan | 100 | 69 | 0,75 |
| sawo | 100 | 92 | 1,75 |

| Makanan Ringan Golongan A | | | |
|---------------------------|-----------|--------|------|
| Nama Masakan | Berat(gr) | Kalori | Unit |
| Arem – Arem | 75 | 225 | 2,75 |
| Bubur Kacang Ijo | 100 | 102 | 1,25 |
| Ketupat Ketan | 120 | 216 | 2,75 |
| Lemper | 70 | 247 | 3 |
| Lepet | 120 | 210 | 2,5 |
| Lepet Ketan | 170 | 346 | 4,25 |

| Makanan Ringan Golongan B | | | |
|---------------------------|-----|-----|------|
| Bolu Gulung | 110 | 300 | 3,75 |
| Cakwee | 50 | 143 | 1,75 |
| Getuk Lindri | 60 | 127 | 1,75 |
| Kerak Telur | 120 | 599 | 7,5 |
| Sandwich | 100 | 164 | 2 |
| Serabi Pandan | 60 | 137 | 1,75 |

| | | | |
|------------------|-----|-----|---|
| Semar Mendem | 100 | 247 | 3 |
| Uli + Tape Ketan | 160 | 559 | 7 |

| Makanan Ringan Golongan C | | | |
|---------------------------|-----|-----|------|
| Bakpia | 25 | 68 | 1 |
| Bakwan | 100 | 270 | 3,25 |
| Bika Ambon | 50 | 99 | 1,25 |
| <i>Black Forrest</i> | 200 | 585 | 7,5 |
| Kastengels (10 bh) | 100 | 426 | 5,75 |
| Keju Lembaran (1bh) | 20 | 65 | 3,25 |
| Kerupuk Mie | 25 | 119 | 1,5 |
| Klepon | 60 | 68 | 0,75 |
| Kroket Kentang | 75 | 146 | 1,25 |
| Kue Ape | 60 | 151 | 2 |
| Kue Cubit | 60 | 183 | 2,25 |
| Kue Cucur | 90 | 152 | 2 |
| Kue Ku | 50 | 237 | 3 |
| Kue Lumpur | 80 | 232 | 3 |
| Kue Pukis | 40 | 181 | 2,25 |
| Lapis Legit | 50 | 307 | 3,75 |

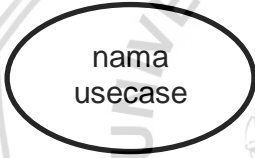


2.6 Use case Diagram

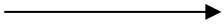

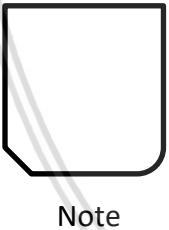
Menurut Sugiarti Fungsi dari *Use case diagram* sendiri ialah untuk memodelkan proses bisnis berdasarkan sudut pandang pengguna sistem. *Use case diagram* terdiri dari dua elemen yaitu *diagram* untuk *use case* dan actor. Actor ialah presentasi orang yang akan mengoperasikan atau orang yang berinteraksi dengan sistem aplikasi. Sedangkan *Use case* merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor (Sugiarti, 2013).

Terdapat beberapa simbol yang digunakan didalam pembuatan *use case diagram*, simbol - simbol yang akan dijelaskan didalam tabel 2.3 berikut

Tabel 2.3 Tabel Simbol Pada Use case Diagram

sumber (Satzinger, et al (2012))

| No | Gambar | Keterangan |
|----|---|--|
| 1 |  | Simbol Oval disamping bernama " Use case ". Kegunaan dari simbol tersebut ialah untuk menandakan unit - unit mana yang saling berinteraksi. Selain itu usecase juga berfungsi untuk memberi informasi aksi apa saja yang bisa dilakukan oleh aktor. Penamaan <i>use case</i> sebaiknya menggunakan kalimat kerja seperti Hapus data, Menambah data pegawai dan lainnya. |
| 2 |  | Simbol disamping bernama Aktor . Fungsi dari simbol tersebut adalah sebagai penanda objek - objek diluar sistem yang berinteraksi dengan sistem yang sedang di bangun. Dalam penamaanya simbol aktor memiliki unsur objek |
| 3 |  | Simbol yang nampak seperti garis lurus disamping bernama Asosiasi . Kegunaan dari simbol disamping adalah untuk menggambarkan hubungan antara unit satu an unit lainnya. Berbeda dengan simbol yang sudah di jelaskan sebelumnya, simbol ini tidak diberikan penamaan hanya di gambar garis lurus biasa didalam <i>diagram</i> usecase |

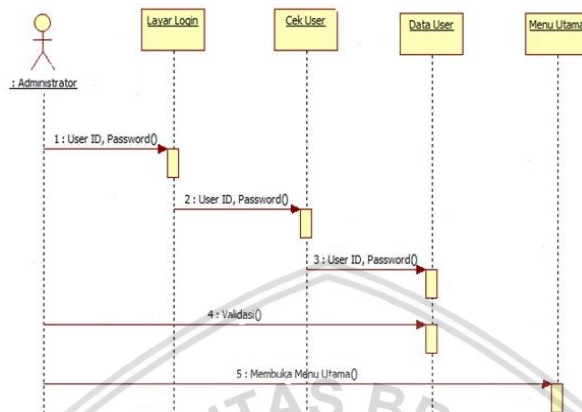
| | | |
|---|---|---|
| 4 | <<extend>> | Extend memiliki fungsi untuk memberikan tanda pada hubungan antara <i>use case</i> dengan <i>use case</i> lainnya, ketika adanya perluasan dari <i>use case</i> lain jika kondisi atau syarat terpenuhi. |
| 5 |  | Generalisasi merupakan nama dari simbol disamping. Generalisasi pada <i>Use case</i> atau aktor dapat diartikan sebagai Use-case atau aktor yang satu dengan lainnya merupakan spesialisasi dari yang lain. |
| 6 | <<include>> | Include digunakan ketika ada indikasi salah satu <i>use case</i> , merupakan bagian <i>use case</i> yang lain (required). dengan kata lain <i>use case</i> tersebut harus dipenuhi untuk menjalankan usecase yang lainnya |
| 7 |  | Menspesifikasikan paket yang menampilkan sistem secara terbatas. |
| 8 |  | Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi |

2.7 Sequence Diagram

Sequence diagram adalah grafik dua dimensi dimana obyek ditunjukkan dalam dimensi horizontal, sedangkan *lifeline* ditunjukkan dalam dimensi vertikal. (Munawar ,2005)

Sequence diagram adalah suatu *diagram* interaksi yang menekankan pada pengaturan waktu dari pesan-pesan. *Diagram* ini menampilkan sekumpulan peran dan pesan-pesan yang dikirim dan diterima oleh instansi yang memegang peranan tersebut. *Sequence diagram* menangkap objek dan *class* yang terlibat dalam Skenario dan urutan pesan yang ditukar antara objek diperlukan untuk melaksanakan fungsionalitas Skenario.

Sequence diagram berasosiasi dengan *use case* selama proses pengembangan. Dalam *Unified Model Language* (UML), objek dalam *sequence diagram* digambar dengan segiempat yang berisi nama objek yang diberi garis bawah. Objek dapat diberi nama dengan tiga cara: (nama objek), (nama objek dan *class*) atau (hanya nama *class* (*anonymous object*)). (Booch ,2005). Berikut gambar 2.1 yang merupakan contoh dari *Sequence Diagram*.



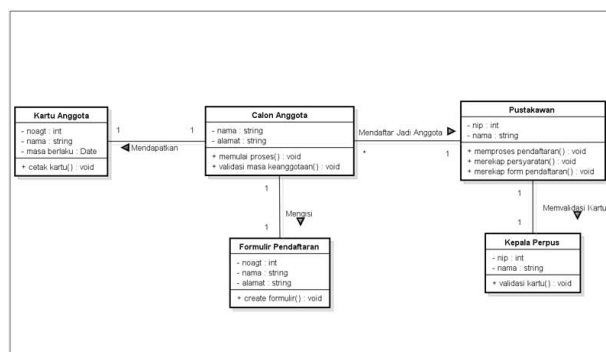
Gambar 2.1 Contoh Sequence Diagram

Sumber (Booch (2005)).

2.8 Class Diagram

Satzinger, et al (2012:101), Menyatakan bahwa *class diagram* adalah *diagram* yang terdiri dari sekumpulan kelas yang saling berhubungan dan asosiasi antar kelas, sedangkan didalam buku *System Analysis & Design Fifth Edition* Dennis, et all (2012:521) menyatakan bahwa *class diagram* adalah sebuah model yang bersifat statis yang dapat mendukung sebuah evolusi dari sistem.

Sebuah *Class* terdapat 3 aelemen utama yaitu: atribut nama *class* dan operasi Atribut memiliki fungsi untuk memberikan sifat / karakteristik data yang terdapat didalam *class* tersebut, nama memiliki fungsi untuk memberikan identitas suatu *class* sedangkan operasi fungsinya adalah memberikan sebuah fungsi ke sebuah objek. Berikut gambar 2.2 yang merupakan contoh dari sebuah *class diagram*



Gambar 2.2 Contoh Class Diagram

Sumber (Satzinger (2012))

2.9 SQLITE

Banyak jenis embedded database yang ada, salah satunya adalah SQLITE. SQLite merupakan sebuah embedded database yang sangat terkenal karena menggabungkan antarmuka SQL dengan memori yang sangat kecil dan kecepatan yang baik. (Gargenta, 2011). SQLite adalah sebuah open *source* database yang telah ada cukup lama, cukup stabil, dan sangat terkenal pada perangkat kecil, termasuk *Android*

Android menyediakan database relasional yang ringan untuk setiap aplikasi menggunakan SQLite. Aplikasi dapat mengambil keuntungan dari itu untuk mengatur relational database engine untuk menyimpan data secara aman dan efisien (Meier, 2010).

2.10 White Box Testing

White-box testing adalah pendekatan dalam pengujian program dimana tes didasarkan pada pengetahuan tentang struktur program dan komponennya (Pressman, 2010). Pada *white-box testing* pengujian didasarkan pada kasus uji yang diperoleh melalui pencarian jalur independen kode program. Jalur independen kode program dapat diketahui melalui perancangan *flow graph*.

Flow graph adalah *diagram* yang menggambarkan alur kode program yang terdiri dari node (simpul) yang biasanya direpresentasikan dalam bulatan berisi data (biasanya angka yang merepresentasikan urutan alur program dan edge yaitu penghubung antar node yang memiliki arah pada satu simpul tertentu.

2.11 Black Box Testing

Black box testing atau pengujian behavioral Menurut Roger S. Pressman (2010) adalah Pengujian yang hanya berfokus kepada fungsional dari perangkat lunak. Dimana pernyataan tersebut dapat diartikan teknik *Black Box Testing* memungkinkan untuk mendapatkan set kondisi masukan yang sepenuhnya akan melaksanakan semua persyaratan fungsional untuk suatu program.

Roger S. Pressman memiliki pendapat *Black Box Testing* bukan merupakan alternatif dari pengujian *White Box Testing*. Sebaliknya, *Black Box Testing* adalah pendekatan komplementer yang mungkin untuk mengungkap kelas yang berbeda dari kesalahan daripada metode *White Box Testing*.

Black Box Testing mencoba untuk menemukan kesalahan dalam kategori berikut.

- Fungsi tidak benar atau hilang.
- Kesalahan *interface* atau antarmuka.
- Kesalahan dalam struktur data atau akses database eksternal.
- Kesalahan kinerja atau perilaku.
- Kesalahan inisialisasi dan terminasi.

2.12 Usability Testing

Usability Testing dilakukan untuk melakukan pengujian sistem dengan cara menguji sistem secara langsung kepada sejumlah responden untuk mengetahui tingkat kemudahan penggunaan sebuah sistem. Pengujian ini dilakukan setelah sistem sudah siap digunakan sehingga diperoleh penilaian berdasarkan hasil dan analisis responden yang dihitung berdasarkan metode yang digunakan. Dalam pengujian usabilitas kali ini digunakan metode *System Usability Scale* (USU)

2.12.1 System Usability Scale

System Usability Scale (SUS) merupakan sebuah pengujian pada sisi pengguna yang menyediakan alat ukur yang quick and dirt dan dapat diandalkan (Brooke, 1986). Pada SUS metode ini menggunakan 10 pernyataan berbentuk kuisioner yang dijawab dengan 5 opsi jawaban untuk setiap pernyataannya, mulai dari Sangat Setuju hingga Sangat Tidak Setuju dengan nilai liker 0 sampai 4. Metode ini pertama kali diperkenalkan oleh John Brooke pada tahun 1986 yang digunakan untuk mengevaluasi berbagai jenis produk ataupun servis, termasuk hardware, software, perangkat mobile, website bahkan aplikasi.

Metode uji pengguna *System Usability Scale* ini menggunakan 10 item pernyataan yang dijelaskan pada tabel 2.4

Tabel 2.4 Tabel Kuisioner *System Usability Scale*

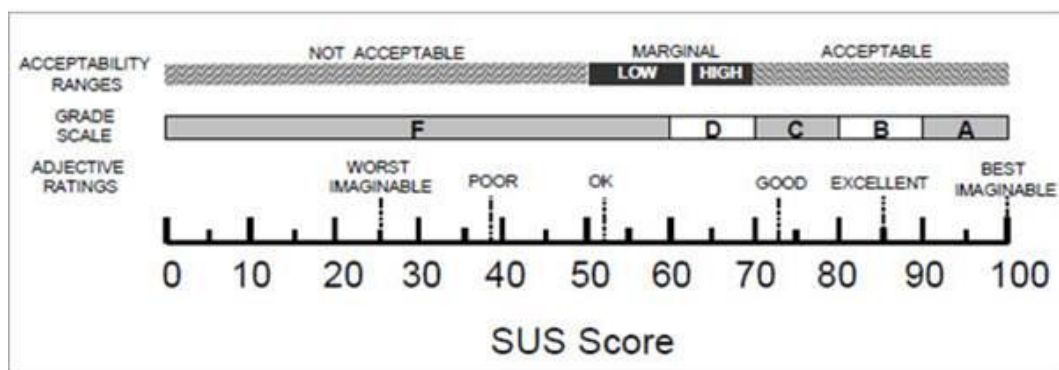
| No | Pertanyaan | Sangat Tidak Setuju | | | | Sangat Setuju |
|----|---|---------------------|---|---|---|---------------|
| | | 1 | 2 | 3 | 4 | 5 |
| 1 | Saya rasa saya akan sering menggunakan sistem ini. | | | | | |
| 2 | Saya merasa sistem terlalu kompleks padahal sebenarnya dapat dibuat sederhana. | | | | | |
| 3 | Saya rasa sistem mudah untuk digunakan. | | | | | |
| 4 | Saya rasa saya membutuhkan bantuan dari orang teknis untuk dapat menggunakan sistem ini. | | | | | |
| 5 | Saya menemukan bahwa terdapat berbagai macam fungsi yang terintegrasi dengan baik dalam sistem. | | | | | |

| | | | | | | |
|----|--|--|--|--|--|--|
| 6 | Saya rasa banyak hal yang tidak konsisten terdapat pada sistem. | | | | | |
| 7 | Saya rasa mayoritas pengguna akan belajar menggunakan sistem ini secara cepat. | | | | | |
| 8 | Saya menemukan bahwa sistem sangat tidak praktis. | | | | | |
| 9 | Saya sangat percaya dalam menggunakan sistem ini. | | | | | |
| 10 | Saya harus belajar banyak hal terlebih dahulu sebelum saya dapat menggunakan sistem ini. | | | | | |

Dari tabel 2.4 tersebut dilakukan sebuah perhitungan, dimana Perhitungan bobot untuk masing masing item mempunyai aturan sebagai berikut:

1. Untuk item no 1,3,5,7, dan 9 nilai yang didapat adalah posisi skala dikurangi 1.
2. Untuk item 2,4,6,8 dan 10, nilai yang didapat adalah 5 dikurangi posisi skala
3. Kemudian jumlahkan nilai yang didapat dan lakukan proses perhitungan nilai jumlah dikalikan 2,5 untuk mendapatkan nilai keseluruhan skor SUS.

SUS (*System Usability Scale*) adalah paket pengujian *usability* yang efektif dan handal untuk digunakan pada berbagai produk dan aplikasi (Bangor, Kortum, & Miller, 2009). Total skor didapatkan dengan menjumlahkan seluruh skor tiap pertanyaan (genap maupun ganjil). Sedangkan skor SUS didapat dengan cara mengkalikan total skor dengan 2.5. Skor akhir SUS akan berada pada kisaran 0-100. Dari hasil skor SUS tersebut dapat kita ketahui seberapa tinggi tingkat *usability* dan akseptabilitas (*acceptable*) desain sistem aplikasi yang dikembangkan. Penilaiannya berdasarkan tiga kategori yaitu *Not Acceptable* atau belum dapat diterima dengan rentang skor SUS 0-50.9, *Marginal* 51-70.9, dan *Acceptable* 71-100. Responden diminta menjawab semua butir pertanyaan yang diisi setelah pengguna selesai menggunakan sistem secara keseluruhan. Gambar 2.3 merupakan gambaran dari skala penilaian dari hasil akhir SUS yang didapat.



Gambar 2.3 Skala Nilai Dari Skor SUS

Sumber (Determining What Individual SUS Scores Mean Adding an Adjective Rating Scale, Kortum, & Miller, 2009)

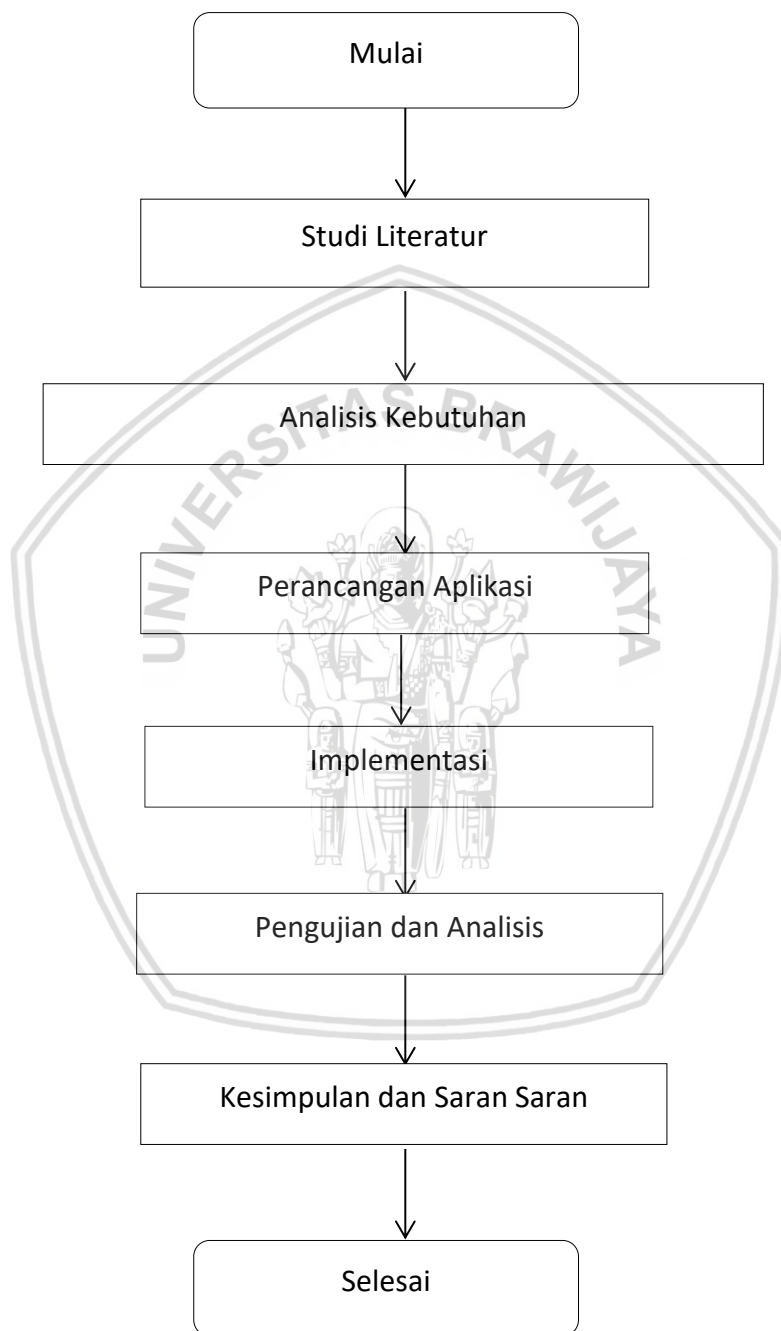
2.13 Konsep MVC

Pengertian dari konsep MVC sendiri ialah sebuah metode yang digunakan untuk membuat sebuah aplikasi dengan memisahkan data atau query (Model) dari tampilan atau *user interface* (View) dan cara pemrosesannya (Controller). Pembuatan program yang menggunakan MVC ini biasanya dikemas dalam kerangka kerja (framework), sehingga pengembang aplikasi tinggal menggunakan kerangka kerja yang sudah disediakan. (Badiyanto, 2013). Terdapat 3 aspek yang terdapat pada MVC yaitu Model, View dan Controller. Berikut adalah pengertian dari ketiga aspek tersebut.

- A. Model
Model merupakan kelas yang mendasari logika proses dalam aplikasi perangkat lunak dan kelas yang terkait dengannya. Model adalah suatu objek yang tidak mengandung informasi tentang *user interface*.
- B. View
View merupakan kumpulan dari kelas yang mewakili unsur-unsur dalam *user interface* (semua hal *user* dapat melihat dan merespon pada layar, seperti tombol, tampilan kotak, dan sebagainya).
- C. Controller
Controller merupakan kelas yang menghubungkan model dan *view*, dan digunakan untuk berkomunikasi antara kelas dalam model dan *view*.

BAB 3 METODE PENELITIAN

Pada bab ini akan dijelaskan langkah – langkah yang akan dilakukan dalam perancangan, implementasi dan pengujian dari aplikasi perangkat lunak ini. Berikut merupakan *diagram* alir urutan pengerjaan penelitian ini:



Gambar 3.1 *Diagram* Alir Pengerjaan Skripsi

3.1 Studi Literatur

Studi literature dilakukan dengan melakukan wawancara terhadap seseorang yang *expert* didalam mengatur menu makanan yang cocok bagi penderita diabetes. Beliau bernama Maria Ika Rosita seoran sarjana lulusan S1 IPB jurusan gizi masyarakat dan sumberdaya keluarga. Selain itu sumber literature dapat berupa buku, jurnal, karya ilmiah dan penelitian lainnya. Hasil dari studi literature yang diharpkan ialah bagaimana implementasi dari metode *harris benedict* untuk menghitung jumlah kalori yang dibutuhkan manusia perharinya.

3.2 Analisis Kebutuhan

Analisis kebutuhan bertujuan untuk mendapatkan semua kebutuhan yang diperlukan dari sistem yang akan dibangun. Analisis kebutuhan dilakukan dengan mengidentifikasi semua kebutuhan sistem dan siapa saja yang terkait dengan sistem tersebut. Metode analisis yang digunakan adalah *Object Oriented Analysis*. Analisa yang dilakukan pada tahapan ini yaitu identifikasi aktor, kebutuhan fungsional, kebutuhan non fungsional, *use case diagram*, *use case* skenario. Secara keseluruhan kebutuhan yang dibutuhkan dalam penelitian ini adalah:

- a. Kebutuhan Fungsional
 1. Sistem mampu mengolah data lalu menyediakan menu makanan yang tepat bagi penderita penyakit diabetes dengan metode harris benedict
 2. Sistem mampu mengkalkulasi berapa jumlah kalori yang dibutuhkan pengguna dengan metode harris benedict
 3. Pengguna mampu melihat *List* ahli gizi yang dapat dikirim pesan untuk melakukan janji konsultasi menu makanan
- b. Kebutuhan nonfungsional
 1. Sistem memiliki tampilan yang mudah dipahami pengguna

3.3 Perancangan Sistem

Pada tahap ini dijelaskan dari desain sistem secara menyeluruh mulai dari model sampai arsitektur yang digunakan dalam pembuatan sistem. Proses perancangan berguna dilakukan untuk mempermudah proses implementasi nantinya. Dimana proses perancangan menjadi media pemandu untuk proses implementasi sehingga hasil dari implementasi yang dilakukan sesuai dengan analisa kebutuhan. Dimana dalam perancangannya aplikasi ini akan memanfaatkan metode model *view* dan controller.

Beberapa tahapan yang dilakukan nantinya didalam perancangan aplikasi yaitu perancangan perhitungan kalori menggunakan metode *harris benedict*, perancangan penentuan menu makanan, pembuatan *sequence diagram*, pembuatan *class diagram*, pembuatan *physical data model* dan perancangan antarmuka.

3.4 Implementasi

Tahap implementasi ini dilakukan dengan menggunakan bahasa java berbasis platform *android* studio dengan pemanfaatan database SQLITE. aplikasi ini memanfaatkan pemodelan OOP. dimana salah satu objek yang ada adalah pengguna. Kemudian aplikasi menerima masukan yang diberikan pengguna berupa variabel – variabel yang nantinya berguna untuk mengkalkulasi menu yang ideal bagi dirinya variable variable tersebut adalah:

- a. Nama pengguna
- b. Tanggal lahir pengguna
- c. Berat badan pengguna
- d. Tinggi badan pengguna
- e. Jenis kelamin

Kemudian variable - variable tersebut diolah menggunakan metode *harris benedict* yang akan memunculkan kalori yang dibutuhkan pengguna perharinya,

Perempuan : $655 + (9,6 \times BB) + (1,8 \times TB) - (4,7 \times U)$

Laki-laki : $66 + (13,7 \times BB) + (5 \times TB) - (6,8 \times U)$

Keterangan :

BB : Berat badan dalam Kg

TB : Tinggi badan dalam Cm

U : Usia dalam Tahun

lalu keluaran kalori tersebut akan disesuaikan dengan kandungan kalori makanan yang ada. sehingga pengguna mendapatkan menu makanan tiap hari yang cocok bagi dirinya.

3.5 Pengujian Aplikasi

Tahapan pengujian dan analisis berfungsi untuk membuat aplikasi yang dibangun agar mampu berkerja sesuai engan spesifikasi kebutuhan dan berjalan sesuai dengan harapan yang diinginkan sebelumnya. terdapat dua macam pengujian yang dilakukan yaitu *black box testing*.

Pengujian *white box testing* yang dilakukan yaitu unit *testing*, dimana unit *testing* dilakukan dengan cara manual dengan cara memverifikasi dan validasi setiap unit pada *Source Code*. Sebuah unit merupakan bagian terkecil dari aplikasi yang dapat di uji coba. Unit adalah fungsi individual atau prosedur.

Sedangkan pengujian *black box testing* yang dilakukan menggunakan metode fungsional *testing*, dengan melakukan verifikasi apakah aplikasi sudah berjalan sesuai dengan kebutuhan fungsional yang sudah dirancang sebelumnya pada tahap disain.

Lalu tahapan pengujian yang terakhir dilakukan adalah pengujian *usability*. Penguian *usability* dilakukan dengan memanfaatkan template kuisisioner dari

System Usability Scale. Responden yang dipilih untuk melakukan pengujian *usability* tersebut merupakan *user* dan ahli gizi. Dimana *user* dipilih dari perawat – perawat panti jompo yang menanganani pemberian asupan makanan bagi individu yang ada di panti jompo Pondok Lansia Al-Ishlah, dimana individu tersebut sudah diketahui memiliki riwayat penyakit diabetes.



BAB 4 ANALISA KEBUTUHAN DAN PERANCANGAN

Perancangan aplikasi yang akan dibangun merupakan hasil dari analisis kebutuhan yang telah dilakukan. Ada beberapa tahapan yang dilakukan didalam di bab ini dimulai dari perancangan aplikasi analisa kebutuhan, pembuatan usecase *diagram*, *use case scenario*, *class diagram* serta perancangan antar muka.

4.1 Analisa Kebutuhan

Penentuan menu makanan bagi penyandang diabetes memerlukan sebuah perhitungan khusus. Dimana tidak semua orang mengetahui ilmu tersebut, ilmu untuk menentukan makanan yang sebanding dengan kalori yang dibutuhkan penyandang diabetes. Sedangkan Jasa untuk melakukan konsultasi dengan ahli gizi yang berpengalaman tidaklah murah, sementara itu banyak lulusan baru ahli gizi yang mampu untuk menentukan menu makanan yang cocok bagi penyandang diabetes tersebut.

Sistem yang sedang dibangun ini berusaha untuk menyelesaikan permasalahan diatas, dengan membuat sistem yang berbasis mobile yang dapat menentukan menu makanan yang tepat bagi penyandang diabetes serta melakukan konsultasi dengan lulusan - lulusan baru ahli gizi terutama di universitas brawijaya. Pada analisa kebutuhan ini dimulai dengan identifikasi aktor yang terlibat dengan sistem serta penjabaran kebutuhan fungsional dan memodelkannya kedalam *use case diagram*.

Semua kebutuhan sistem aplikasi dan siapa saja aktor yang terlibat di dalam aplikasi nantinya dimodelkan dalam *diagram use case*. Tiap *use case* tersebut dijelaskan lebih detailnya didalam Skenario *use case*.

4.1.1 Identifikasi Aktor

Pada penelitian ini terdapat dua aktor yang akan berinteraksi dengan sistem yaitu pengguna (pengguna) dan ahli gizi. Pengguna dapat menghitung kalori yang ia butuhkan dan melakukan konsultasi dengan ahli gizi nantinya. Seangkan ahli gizi dapat melihat permintaan konsultasi dari pengguna serta melihat data pribadi dari pengguna dan menentukan menu makanan mana yang tepat untuk pengguna tersebut.

Tabel 4.1 Tabel Identifikasi Aktor

| Aktor | Deskripsi Aktor |
|-----------|---|
| Pengguna | Aktor yang menggunakan aplikasi untuk menentukan menu makanan mana yang tepat bagi dirinya atau melakukan konsultasi dengan ahli gizi. |
| Ahli Gizi | Aktor yang menjadi konsultan ahli gizi dan menerima permintaan konsultasi bagi pengguna dan menentukan menu makanan yang tepat bagi pengguna. |

4.1.2 Identifikasi Kebutuhan

Identifikasi Kebutuhan merupakan sebuah tahapan untuk mengumpulkan seluruh kebutuhan dalam pembuatan sistem. Identifikasi kebutuhan terdiri dari kebutuhan fungsional dan kebutuhan non fungsional.

1. Kebutuhan Fungsional

Pada daftar kebutuhan fungsional akan dispesifikasikan yaitu spesifikasi kebutuhan fungsional pengguna yang ditunjukkan pada tabel 4.2 dengan penomoran menggunakan FR (*Functional Requirement*).

Tabel 4.2 Tabel Kebutuhan Fungsional

| Nomor FR | Kebutuhan | Usecase | Aktor |
|----------|--|------------------------|----------|
| FR_1 | Aplikasi mampu menyediakan fungsi untuk melakukan register dengan inputan data pribadi pengguna. | Melakukan Registrasi | Pengguna |
| FR_2 | Aplikasi mampu menyediakan fungsi untuk melakukan <i>login</i> berdasarkan proses registrasi yang telah dilakukan pengguna sebelumnya. | Melakukan <i>Login</i> | Pengguna |
| FR_3 | Aplikasi mampu menyediakan fungsi untuk menambahkan biodata pengguna. | Menambahkan Biodata | Pengguna |
| FR_4 | Aplikasi mampu menyediakan fungsi untuk memperbarui biodata pengguna. | Memperbarui Biodata | Pengguna |
| FR_5 | Aplikasi mampu menyediakan fungsi untuk memperbarui profil pengguna. | Memperbarui Profil | Pengguna |

| | | | |
|-------|---|-------------------------------------|-----------|
| FR_6 | Aplikasi mampu melakukan perhitungan kalori yang dibutuhkan oleh pengguna sesuai metode <i>harris benedict</i> . | Menghitung Kebutuhan Kalori perhari | Pengguna |
| FR_7 | Aplikasi mampu menampilkan saran menu makanan bagi pengguna per minggunya sesuai dengan kalori yang dibutuhkan. | Melihat Saran Menu Makanan | Pengguna |
| FR_8 | Aplikasi mampu menampilkan <i>List</i> ahli gizi yang dapat dijadikan konsultan makanan bagi penderita diabetes. | Melihat <i>List</i> Ahli Gizi | Pengguna |
| FR_9 | Aplikasi menyediakan fungsi untuk mengirim pesan dari pengguna untuk ahli gizi yang telah dipilih oleh pengguna sebagai konsultan menu makanan. | Mengirim Pesan ke Ahli Gizi | Pengguna |
| FR_10 | Aplikasi menyediakan fungsi untuk membalas pesan dari ahli gizi untuk pengguna. | Membalas Pesan ke Ahli Gizi | Pengguna |
| FR_11 | Aplikasi menyediakan fungsi untuk <i>logout</i> dari akun pengguna. | Melakukan <i>Logout</i> | Pengguna |
| FR_12 | Aplikasi menyediakan fungsi untuk melakukan register dengan | Melakukan Registrasi Ahli Gizi | Ahli Gizi |

| | | | |
|-------|---|-------------------------------------|-----------|
| | inputan data pribadi ahli gizi. | | |
| FR_13 | Aplikasi mampu menyediakan fungsi untuk melakukan <i>login</i> berdasarkan proses registrasi yang telah dilakukan ahli gizi sebelumnya. | Melakukan <i>Login</i> Ahli Gizi | Ahli Gizi |
| FR_14 | Aplikasi mampu menyediakan fungsi untuk menambahkan biodata ahli gizi. | Menambahkan Biodata Ahli Gizi | Ahli Gizi |
| FR_15 | Aplikasi mampu menyediakan fungsi untuk memperbarui biodata ahli gizi. | Memperbarui Biodata Ahli Gizi | Ahli Gizi |
| FR_16 | Aplikasi mampu menyediakan fungsi untuk membalas pesan yang dikirim pengguna ke ahli gizi. | Membalas Pesan ke Pengguna | Ahli Gizi |
| FR_17 | Aplikasi mampu melakukan perhitungan kalori yang dibutuhkan oleh pengguna sesuai metode <i>harris benedict</i> . | Menghitung Kebutuhan Kalori perhari | Ahli Gizi |
| FR_18 | Aplikasi mampu menyediakan fungsi untuk <i>logout</i> dari akun ahli gizi. | Melakukan <i>Logout</i> Ahli Gizi | Ahli Gizi |

2. Kebutuhan Non Fungsional

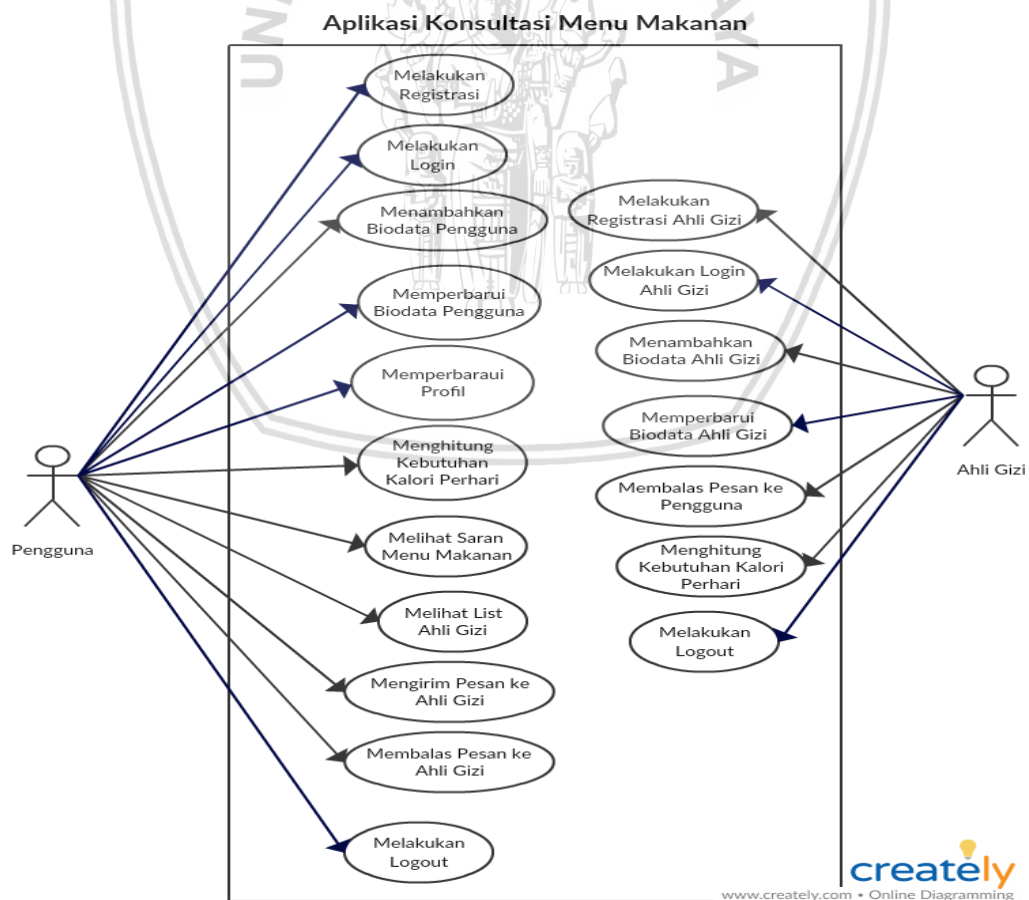
Pada daftar kebutuhan non fungsional akan dispesifikasikan yaitu spesifikasi kebutuhan non fungsional sistem yang ditunjukkan pada tabel 4.3 dengan penomoran menggunakan NFR (NonFunctional Requirement Specification).

Tabel 4.3 Tabel Kebutuhan Non Fungsional

| Nomor NFR | Kebutuhan | Parameter |
|-----------|---|------------------|
| NFR_1 | Aplikasi memiliki tampilan yang mudah dipahami pengguna | <i>Usability</i> |

4.1.3 Use case Diagram

Use case Diagram pada sistem ini melibatkan dua aktor yang terlibat, aktor tersebut ialah pengguna dan ahli gizi. Gambar 4.1 dibawah ini menunjukkan *Use case Diagram* dari aplikasi konsultasi menu makanan bagi penderita diabetes menggunakan metode *harris benedict*.



Gambar 4.1 Use case Diagram Aplikasi

4.1.4 Use case Skenario

Pada Skenario *use case* ini akan dibahas secara lebih detail mengenai usecase *diagram* yang dibuat sebelumnya. Penjabaran yang dilakukan pada *use case* Skenario ini mencakup beberapa hal, yaitu nama *use case*, tujuan *use case* yang terdapat didalam sistem serta penjelasan tahap demi tahap tanggapan sistem terhadap aksi yang diberikan oleh aktor

a. Use case Skenario Melakukan Registrasi

Pada *use case* skenario melakukan registrasi akan dijelaskan lebih detail mengenai proses seorang pengguna melakukan registrasi terhadap aplikasi. *Use case* melakukan registrasi dilakukan untuk mengumpulkan data yang diperlukan nantinya dalam perhitungan kalori menggunakan metode *harris benedict*. Tabel 4.4 dibawah ini merupakan *use case* Skenario melakukan registrasi.

Tabel 4.4 Use case Skenario Melakukan Registrasi

| | |
|---------------|---|
| Nama Use case | Melakukan Registrasi |
| Aktor | Pengguna |
| Tujuan | Melakukan registrasi aplikasi serta memberikan data pribadi pengguna |
| Deskripsi | <i>Use case</i> ini digunakan sebagai registrasi awal pengguna sehingga sistem mengetahui data - data pribadi pengguna yang dapat berguna bagi proses lainnya yang ada di aplikasi |
| Kondisi Awal | Aktor telah membuka aplikasi serta menyiapkan data pribadinya |
| Kondisi Akhir | Aplikasi telah berhasil melakukan registrasi dan pengguna diarahkan menuju tampilan utama |
| Main Flow | <ol style="list-style-type: none"> 1. Aktor memilih pilihan pengguna pada <i>splashscreen</i> aplikasi. 2. Aktor memilih pilihan registrasi di aplikasi. 3. Aplikasi menampilkan <i>form</i> registrasi. 4. Aktor memasukan data pribadi sesuai dengan <i>form</i> yang telah tersedia. |

| | |
|------------------|--|
| | <ol style="list-style-type: none"> 5. aktor memilih tombol register. 6. aplikasi berhasil menyimpan data aktor dan melakukan registrasi aplikasi menampilkan menu utama. |
| Alternative Flow | |

b. *Use case* Skenario Melakukan *Login*

Pada *use case* skenario Melakukan *Login* akan dijelaskan lebih detail mengenai proses seorang pengguna melakukan *login* aplikasi. *Use case* ini berfungsi bagi pengguna untuk masuk kedalam aplikasi. Tabel 4.5 dibawah ini merupakan *use case* skenario Melakukan *Login*.

Tabel 4.5 *Use case* Skenario Melakukan *Login*

| | |
|----------------------|---|
| Nama <i>Use case</i> | Melakukan <i>Login</i> |
| Aktor | Pengguna |
| Tujuan | Masuk kedalam aplikasi sebagai pengguna |
| Deskripsi | <i>Use case</i> ini digunakan pengguna untuk masuk kedalam aplikasi |
| Kondisi Awal | Aktor telah berhasil registrasi |
| Kondisi Akhir | Aktor telah berhasil masuk kedalam halaman beranda aplikasi |
| Main Flow | <ol style="list-style-type: none"> 1. Aktor memilih pilihan pengguna pada <i>splashscreen</i> aplikasi. 2. Aktor memilih pilihan <i>login</i> pada aplikasi. 3. Aktor mengisi <i>form login</i> yang berisi <i>username</i> dan <i>password</i>, sesuai dengan data registrasi sebelumnya. 4. Aktor memilih pilihan <i>login</i>. 5. Aktor berhasil masuk kedalam halaman beranda aplikasi |
| Alternative Flow | <ol style="list-style-type: none"> 1. Aktor memilih pilihan pengguna pada <i>splashscreen</i> aplikasi. 2. Aktor memilih pilihan <i>login</i> pada aplikasi. |

| | |
|--|---|
| | <ol style="list-style-type: none"> Aktor mengisi <i>form login</i> yang berisi <i>username</i> dan <i>password</i>, dengan data yang salah. Aplikasi menampilkan pemberitahuan bahwa data yang diberikan salah. |
|--|---|

c. *Use case* Skenario Menambahkan Biodata Pengguna

Pada *use case* Skenario Menambahkan Biodata Pengguna akan dijelaskan lebih detail mengenai proses seorang pengguna menambahkan data biodata ke aplikasi. *Use case* ini berfungsi untuk menambahkan data biodata pengguna ke aplikasi. Tabel 4.6 dibawah ini merupakan *use case* Skenario Menambahkan Biodata Pengguna.

Tabel 4.6 *Use case* Skenario Menambahkan Biodata Pengguna

| | |
|----------------------|---|
| Nama <i>Use case</i> | Menambahkan Biodata Pengguna |
| Aktor | Pengguna |
| Tujuan | Menambahkan data dari biodata pengguna ke aplikasi |
| Deskripsi | <i>Use case</i> ini digunakan sebagai penambahan data biodata pengguna setelah melakukan registrasi email dan nama |
| Kondisi Awal | Aktor telah berhasil registrasi |
| Kondisi Akhir | Data biodata berhasil disimpan didalam database |
| Main Flow | <ol style="list-style-type: none"> Aktor berhasil registrasi email dan password Aktor menekan tombol registrasi Aktor mengisi <i>form</i> biodata Aktor menekan tombol simpan Sistem menyimpan data yang telah diperbarui kedalam database |
| Alternative Flow | |

d. *Use case* Skenario Memperbarui Biodata Pengguna

Pada *use case* Skenario Menambahkan Biodata Pengguna akan dijelaskan lebih detail mengenai proses seorang pengguna memperbarui biodata dari akun pengguna di aplikasi. *Use case* ini berfungsi untuk memperbarui biodata pengguna dari aplikasi. Tabel 4.7 dibawah ini merupakan *use case* Skenario Memperbarui Biodata Pengguna.

Tabel 4.7 *Use case* Skenario Memperbarui Biodata Pengguna

| | |
|----------------------|---|
| Nama <i>Use case</i> | Memperbarui Biodata Pengguna |
| Aktor | Pengguna |
| Tujuan | Memperbarui biodata pengguna dari akun pengguna di aplikasi |
| Deskripsi | <i>Use case</i> ini digunakan pengguna untuk meperbarui biodata yang terdapat pada aplikasi |
| Kondisi Awal | Aktor telah berhasil <i>Login</i> |
| Kondisi Akhir | Data biodata berhasil disimpan didalam database |
| Main Flow | <ol style="list-style-type: none"> 1. Aktor memilih pilihan untuk melihat profil pada aplikasi 2. Aktor memilih pilihan untuk mengedit profil. 3. Aktor mengisi <i>form</i> edit profil, dimana didalam <i>form</i> tersebut terdapat <i>form</i> biodata untuk diisi. 4. Aktor menekan tombol simpan. 5. Sistem menyimpan data yang telah diperbarui kedalam <i>database</i>. |
| Alternative Flow | |

e. *Use case* Skenario Memperbarui Profil Pengguna

Pada *use case* Skenario Menambahkan Profil Pengguna akan dijelaskan lebih detail mengenai proses seorang pengguna memperbarui profil dari akun pengguna di aplikasi. *Use case* ini berfungsi untuk memperbarui profil dari akun pengguna di aplikasi. Tabel 4.8 dibawah ini merupakan *a. Use case* Skenario Memperbarui Profil Pengguna.

Tabel 4.8 Use case Skenario Memperbarui Profil Pengguna

| | |
|----------------------|--|
| Nama <i>Use case</i> | Memperbarui Profil Pengguna |
| Aktor | Pengguna |
| Tujuan | Memperbarui profil pengguna pada aplikasi |
| Deskripsi | <i>Use case</i> ini digunakan pengguna untuk meperbarui biodata yang terdapat pada aplikasi |
| Kondisi Awal | Aktor telah berhasil <i>Login</i> |
| Kondisi Akhir | Data biodata berhasil disimpan didalam database |
| Main Flow | <ol style="list-style-type: none"> 1. Aktor memilih pilihan untuk melihat profil pada aplikasi 2. Aktor memilih pilihan untuk mengedit profil. 3. Aktor mengisi <i>form</i> edit profil, dimana didalam <i>form</i> tersebut terdapat <i>form</i> profil untuk diisi. 4. Aktor menekan tombol simpan. 5. Sistem menyimpan data yang telah diperbarui kedalam <i>database</i>. |
| Alternative Flow | |

f. *Use case* Skenario Menghitung Kebutuhan Kalori perhari

Pada *use case* Skenario menghitung kalori perhari akan dijelaskan lebih detail mengenai proses seorang pengguna dapat menghitung kalori yang dibutuhkan tubuhnya perhari dengan aplikasi, dimana rincian perhitungannya akan dijelaskan di bab selanjutnya Tabel 4.9 dibawah ini merupakan *use case* Skenario Menghitung Kalori perhari.

Tabel 4.9 Use case Skenario Menghitung Kebutuhan Kalori perhari

| | |
|----------------------|--|
| Nama <i>Use case</i> | Menghitung Kebutuhan Kalori perhari |
| Aktor | Pengguna |
| Tujuan | Menghitung kalori yang dibutuhkan pengguna perhari |

| | |
|------------------|--|
| Deskripsi | <i>Use case</i> ini berfungsi bagi pengguna untuk menentukan kalori yang dia butuhkan perhari dengan metode <i>harris benedict</i> |
| Kondisi Awal | Aktor telah berhasil <i>login</i> dan masuk ke halaman utama |
| Kondisi Akhir | Aplikasi menampilkan jumlah kalori sesuai dengan metode <i>harris benedict</i> |
| Main Flow | <ol style="list-style-type: none"> 1. Aktor memilih pilihan hitung kalori 2. Aktor mengisi form hitung kalori 3. Aktor memilih pilhan hitung 4. Aplikasi menghitung kalori yang dibutuhkan pengguna sesuai data yang diregistrasi menggunakan metode <i>harris benedict</i>. 5. Aplikasi Menampilkan kalori yang dibutuhkan pengguna. |
| Alternative Flow | |

g. *Use case* Skenario Melihat Saran Menu Makanan

Pada *use case* Skenario melihat saran menu makanan akan dijelaskan lebih detail bagaimana proses seorang pengguna mendapatkan saran menu makanan yang sesuai dengan kalori yang ia butuhkan dengan metode *harris benedict* Tabel 4.10 dibawah ini merupakan *use case* Skenario Menghitung Kalori perhari.

Tabel 4.10 *Use case* Skenario Menampilkan Saran Menu Makanan

| | |
|----------------------|--|
| Nama <i>Use case</i> | Menampilkan Saran Menu Makanan |
| Aktor | Pengguna |
| Tujuan | Melihat saran menu makanan yang sesuai dengan kalori yang dibuthkan pengguna |
| Deskripsi | <i>Use case</i> ini berfungsi bagi pengguna untuk mendapatkan saran menu makanan yang sesuai dengan kalori yang telah dihitung |

| | |
|------------------|---|
| | sebelumnya. Pengguna akan mendapatkan saran menu makanan selama seminggu. |
| Kondisi Awal | Aktor telah berhasil <i>login</i> dan melakukan perhitungan kebutuhan kalori |
| Kondisi Akhir | Aplikasi menampilkan saran menu makanan yang sesuai dengan kalori pengguna |
| Main Flow | <ol style="list-style-type: none"> 1. Aktor memilih pilihan profil pada aplikasi 2. Aplikasi menampilkan halaman profil serta menu makanan yang sesuai dengan kebutuhan kalori pengguna |
| Alternative Flow | |

h. *Use case* Skenario Melihat *List* Ahli Gizi

Pada *use case* Skenario melihat *List* ahli gizi akan dijelaskan lebih detail bagaimana proses seorang pengguna dapat menampilkan *List* ahli gizi yang telah melakukan registrasi dengan aplikasi. Tabel 4.11 dibawah ini merupakan *use case* skenario Melihat *List* Ahli Gizi.

Tabel 4.11 *Use case* Skenario Melihat *List* Ahli Gizi

| | |
|----------------------|---|
| Nama <i>Use case</i> | Melihat <i>List</i> Ahli Gizi |
| Aktor | Pengguna |
| Tujuan | Melihat <i>List</i> ahli gizi yang nantinya dapat diajak konsultasi oleh pengguna |
| Deskripsi | <i>Use case</i> ini berfungsi bagi pengguna untuk melihat <i>List</i> ahli gizi yang dapat mereka pilih sebagai konsultan makanan nantinya |
| Kondisi Awal | Aktor telah berhasil <i>login</i> dan masuk ke halaman utama |
| Kondisi Akhir | Aplikasi menampilkan <i>List</i> ahli gizi terhadap pengguna |
| Main Flow | <ol style="list-style-type: none"> 1. Aktor memilih pilihan menu beranda pada aplikasi. 2. Aplikasi menampilkan <i>List</i> ahli gizi |

| | |
|------------------|--|
| | yang berisi biodata singkat beliau sesuai database aplikasi. |
| Alternative Flow | |

i. *Use case* Skenario Mengirim Pesan ke Ahli Gizi

Pada *use case* Skenario Mengirim Pesan ke Ahli Gizi akan dijelaskan lebih detail bagaimana proses seorang pengguna dapat mengirim pesan kepada ahli gizi yang telah diinginkan menjadi konsultan menu makanan bagi pengguna. Tabel 4.12 dibawah ini merupakan *use case* Skenario Mengirim Pesan ke Ahli Gizi.

Tabel 4.12 *Use case* Skenario Mengirim Pesan ke Ahli Gizi

| | |
|----------------------|---|
| Nama <i>Use case</i> | Melakukan Janji Konsultasi |
| Aktor | Pengguna |
| Tujuan | Pengguna dapat mengirim pesan ke ahli gizi |
| Deskripsi | <i>Use case</i> ini berfungsi bagi pengguna untuk mengirim pesan ke ahli gizi yang diinginkan untuk menjadi konsultan menu makanan |
| Kondisi Awal | Aktor telah berhasil <i>login</i> dan masuk ke halaman utama |
| Kondisi Akhir | Aplikasi berhasil mengirim pesan kepada ahli gizi yang dituju |
| Main Flow | <ol style="list-style-type: none"> 1. Aktor memilih pilihan pesan 2. Aktor memilih pilihan tulis pesan 3. Aplikasi menampilkan <i>form</i> pesan bagi pengguna 4. Aktor menuliskan <i>username</i> ahli gizi yang dapat dilihat di <i>List</i> ahli gizi 5. Aktor menuliskan pesan di <i>form</i> pesan yang disediakan 6. Aktor memilih pilihan kirim yang terdapat didalam <i>form</i> 7. Aplikasi mengirim pesan ke ahli gizi dituju yang ditulis oleh pengguna |
| Alternative Flow | |

j. *Use case* Skenario Membalas Pesan ke Ahli Gizi

Pada *use case* Skenario Membalas Pesan akan dijelaskan lebih detail bagaimana proses seorang pengguna yang dikirim pesan oleh ahli gizi dapat mengirim pesan balasan kepada ahli gizi tersebut. Tabel 4.13 dibawah ini merupakan *use case* Skenario Membalas Pesan ke Ahli Gizi.

Tabel 4.13 *Use case* Skenario Membalas Pesan ke Ahli Gizi

| | |
|----------------------|--|
| Nama <i>Use case</i> | Membalas Pesan ke Ahli Gizi |
| Aktor | Pengguna |
| Tujuan | Pengguna dapat membalas pesan yang dikirim oleh ahli gizi |
| Deskripsi | <i>Use case</i> ini berfungsi bagi ahli gizi untuk membalas pesan yang dikirim oleh pengguna |
| Kondisi Awal | Aktor telah berhasil <i>login</i> dan masuk ke halaman utama |
| Kondisi Akhir | Aplikasi berhasil membalas pesan kepada pengguna yang dituju |
| Main Flow | <ol style="list-style-type: none"> 1. Aktor memilih pilihan pesan 2. aplikasi menampilkan <i>List</i> pesan yang diterima oleh aktor 3. Aktor memilih pesan yang akan dibalas 4. aktor memilih tombol balas yang ada disamping <i>List</i> pesan 5. aplikasi menampilkan <i>form</i> pesan yang digunakan aktor untuk menulis pesan 6. aktor menulis pesan balasan 7. aplikasi mengirim pesan kepada pengguna yang dituju sesuai dengan pengirimnya |
| Alternative Flow | |

k. *Use case* Skenario Melakukan *Logout*

Pada *use case* Skenario melakukan *logout* akan dijelaskan lebih detail bagaimana proses seorang pengguna melakukan *logout* terhadap aplikasi. Tabel 4.14 dibawah ini merupakan *use case* Skenario melakukan *logout* yang dilakukan oleh pengguna.

Tabel 4.14 Use case Skenario Melakukan Logout

| | |
|----------------------|--|
| Nama <i>Use case</i> | Melakukan <i>Logout</i> |
| Aktor | Pengguna |
| Tujuan | Pengguna dapat melakukan <i>logout</i> terhadap aplikasi |
| Deskripsi | <i>Use case</i> ini berfungsi bagi pengguna untuk melakukan <i>logout</i> . |
| Kondisi Awal | Aktor telah berhasil <i>login</i> dan masuk ke halaman utama |
| Kondisi Akhir | Aplikasi berhasil <i>logout</i> dari akun aktor |
| Main Flow | <ol style="list-style-type: none"> 1. Aktor memilih pilihan <i>logout</i> pada aplikasi 2. Aplikasi berhasil <i>logout</i> dari akun aktor 3. Aplikasi menampilkan halaman <i>login</i> |
| Alternative Flow | |

I. *Use case* Skenario Melakukan Registrasi Ahli Gizi

Pada *use case* skenario melakukan registrasi akan dijelaskan lebih detail mengenai proses seorang ahli gizi melakukan registrasi terhadap aplikasi. *Use case* melakukan registrasi dilakukan untuk mengumpulkan data yang diperlukan nantinya dalam menampilkan *list* ahli gizi. Tabel 4.15 dibawah ini merupakan *use case* Skenario Melakukan Registrasi.

Tabel 4.15 Use case Skenario Melakukan Registrasi Ahli Gizi

| | |
|----------------------|---|
| Nama <i>Use case</i> | Melakukan Registrasi |
| Aktor | Ahli gizi |
| Tujuan | Melakukan registrasi aplikasi serta memberikan data pribadi aktor |
| Deskripsi | <i>Use case</i> ini digunakan sebagai registrasi awal ahli gizi sehingga sistem mengetahui data - data pribadi pengguna yang dapat berguna bagi proses lainnya yang ada di aplikasi |
| Kondisi Awal | Aktor telah membuka aplikasi serta menyiapkan data pribadinya |

| | |
|------------------|---|
| Kondisi Akhir | Aplikasi telah berhasil melakukan registrasi dan aktor diarahkan menuju tampilan utama |
| Main Flow | <ol style="list-style-type: none"> 1. Aktor memilih pilihan ahli gizi pada <i>splashscreen</i> aplikasi. 2. Aktor memilih pilihan registrasi di aplikasi. 3. Aplikasi menampilkan <i>form</i> registrasi. 4. Aktor memasukkan data pribadi sesuai dengan <i>form</i> yang telah tersedia. 5. aktor memilih tombol register. 6. aplikasi berhasil menyimpan data aktor dan melakukan registrasi aplikasi menampilkan menu utama. |
| Alternative Flow | |

m. *Use case* Skenario Melakukan *Login* Ahli Gizi

Pada *use case* skenario Melakukan *Login* Ahli Gizi akan dijelaskan lebih detail mengenai proses seorang pengguna melakukan *login* aplikasi. *Use case* ini berfungsi bagi ahli gizi untuk masuk kedalam aplikasi. Tabel 4.16 dibawah ini merupakan *use case* skenario Melakukan *Login* Ahli Gizi.

Tabel 4.16 *Use case* Skenario Melakukan *Login* Ahli Gizi

| | |
|----------------------|--|
| Nama <i>Use case</i> | Melakukan <i>Login</i> Ahli Gizi |
| Aktor | Ahli Gizi |
| Tujuan | Masuk kedalam aplikasi sebagai ahli gizi |
| Deskripsi | <i>Use case</i> ini digunakan ahli gizi untuk masuk kedalam aplikasi |
| Kondisi Awal | Aktor telah berhasil registrasi |
| Kondisi Akhir | Aktor telah berhasil masuk kedalam halaman beranda aplikasi |
| Main Flow | <ol style="list-style-type: none"> 1. Aktor memilih pilihan ahli gizi pada <i>splashscreen</i> aplikasi. 2. Aktor memilih pilihan <i>login</i> pada aplikasi. 3. Aktor mengisi <i>form login</i> yang |

| | |
|------------------|--|
| | <p>berisi <i>username</i> dan <i>password</i>, sesuai dengan data registrasi sebelumnya.</p> <ol style="list-style-type: none"> Aktor memilih pilihan <i>login</i>. Aktor berhasil masuk kedalam halaman beranda aplikasi |
| Alternative Flow | <ol style="list-style-type: none"> Aktor memilih pilihan ahli gizi pada <i>splashscreen</i> aplikasi. Aktor memilih pilihan <i>login</i> pada aplikasi. Aktor mengisi <i>form login</i> yang berisi <i>username</i> dan <i>password</i>, dengan data yang salah. Aplikasi menampilkan pemberitahuan bahwa data yang diberikan salah. |

n. *Use case* Skenario Menambahkan Biodata Ahli Gizi

Pada *use case* Skenario Menambahkan Biodata Ahli Gizi akan dijelaskan lebih detail mengenai proses seorang Ahli Gizi menambahkan data biodata ke aplikasi. *Use case* ini berfungsi untuk menambahkan data biodata ahli gizi ke aplikasi. Tabel 4.17 dibawah ini merupakan *use case* Skenario Menambahkan Biodata Ahli Gizi.

Tabel 4.17 *Use case* Skenario Menambahkan Biodata Ahli Gizi

| | |
|----------------------|---|
| Nama <i>Use case</i> | Menambahkan Biodata Ahli Gizi |
| Aktor | Ahli gizi |
| Tujuan | Menambahkan data dari biodata ahli gizi ke aplikasi |
| Deskripsi | <i>Use case</i> ini digunakan sebagai penambahan data biodata ahli gizi setelah melakukan registrasi email dan nama |
| Kondisi Awal | Aktor telah berhasil registrasi |
| Kondisi Akhir | Data biodata berhasil disimpan didalam database |
| Main Flow | <ol style="list-style-type: none"> Aktor berhasil registrasi email dan password Aktor menekan tombol registrasi |

| | |
|------------------|---|
| | <ol style="list-style-type: none"> 3. Aktor mengisi <i>form</i> biodata 4. Aktor menekan tombol simpan 5. Sistem menyimpan data yang telah diperbarui kedalam database |
| Alternative Flow | |

o. *Use case* Skenario Memperbarui Biodata Ahli Gizi

Pada *use case Skenario* Menambahkan Biodata Ahli Gizi akan dijelaskan lebih detail mengenai proses seorang ahli gizi memperbarui biodata dari akun ahli gizi di aplikasi. *Use case* ini berfungsi untuk memperbarui biodata ahli gizi dari aplikasi. Tabel 4.18 dibawah ini merupakan *use case* Skenario Memperbarui Biodata Ahli Gizi.

Tabel 4.18 *Use case* Skenario Memperbarui Biodata Ahli Gizi

| | |
|----------------------|---|
| Nama <i>Use case</i> | Memperbarui Biodata Ahli Gizi |
| Aktor | Ahli gizi |
| Tujuan | Memperbarui biodata pengguna dari akun pengguna di aplikasi |
| Deskripsi | <i>Use case</i> ini digunakan ahli gizi untuk meperbarui biodata yang terdapat pada aplikasi |
| Kondisi Awal | Aktor telah berhasil <i>Login</i> |
| Kondisi Akhir | Data biodata berhasil disimpan didalam database |
| Main Flow | <ol style="list-style-type: none"> 1. Aktor memilih pilihan untuk melihat profil pada aplikasi 2. Aktor memilih pilihan untuk mengedit profil. 3. Aktor mengisi <i>form</i> edit profil, dimana didalam <i>form</i> tersebut terdapat <i>form</i> biodata untuk diisi. 4. Aktor menekan tombol simpan. 5. Sistem menyimpan data yang telah diperbarui kedalam <i>database</i>. |
| Alternative Flow | |

p. *Use case* Skenario Memperbarui Profil Ahli Gizi

Pada *use case* Skenario Menambahkan Profil Ahli Gizi akan dijelaskan lebih detail mengenai proses seorang ahli gizi memperbarui profil dari pada aplikasi. *Use case* ini berfungsi untuk memperbarui profil dari akun ahli gizi di aplikasi. Tabel 4.19 dibawah ini merupakan *Use case* Skenario *Memperbarui* Profil Ahli Gizi.

Tabel 4.19 *Use case* Skenario Memperbarui Profil Ahli Gizi

| | |
|----------------------|--|
| Nama <i>Use case</i> | Memperbarui Profil Pengguna |
| Aktor | Pengguna |
| Tujuan | Memperbarui profil pengguna pada aplikasi |
| Deskripsi | <i>Use case</i> ini digunakan pengguna untuk meperbarui biodata yang terdapat pada aplikasi |
| Kondisi Awal | Aktor telah berhasil <i>Login</i> |
| Kondisi Akhir | Data biodata berhasil disimpan didalam database |
| Main Flow | <ol style="list-style-type: none"> 1. Aktor memilih pilihan untuk melihat profil pada aplikasi 2. Aktor memilih pilihan untuk mengedit profil. 3. Aktor mengisi <i>form</i> edit profil, dimana didalam <i>form</i> tersebut terdapat <i>form</i> profil untuk diisi. 4. Aktor menekan tombol simpan. 5. Sistem menyimpan data yang telah diperbarui kedalam <i>database</i>. |
| Alternative Flow | |

q. *Use case* Skenario Menghitung Kebutuhan Kalori perhari

Pada *use case* Skenario menghitung kalori perhari akan dijelaskan lebih detail mengenai proses seorang pengguna dapat menghitung kalori yang dibutuhkan tubuhnya perhari dengan aplikasi, dimana rincian perhitungannya akan dijelaskan di bab selanjutnya Tabel 4.9 dibawah ini merupakan *use case* Skenario Menghitung Kalori perhari.

Tabel 4.20 Use case Skenario Menghitung Kebutuhan Kalori perhari

| | |
|----------------------|---|
| Nama <i>Use case</i> | Menghitung Kebutuhan Kalori perhari |
| Aktor | Pengguna |
| Tujuan | Menghitung kalori yang dibutuhkan pengguna perhari |
| Deskripsi | <i>Use case</i> ini berfungsi membantu ahli gizi untuk menghitung kebutuhan kalori perhari yang dibutuhkan seseorang dengan metode <i>harris benedict</i> |
| Kondisi Awal | Aktor telah berhasil <i>login</i> dan masuk ke halaman utama |
| Kondisi Akhir | Aplikasi menampilkan jumlah kalori sesuai dengan metode <i>harris benedict</i> |
| Main Flow | <ol style="list-style-type: none"> 1. Aktor memilih pilihan hitung kalori 2. Aplikasi menghitung kalori yang dibutuhkan pengguna sesuai data yang diregistrasi menggunakan metode <i>harris benedict</i>. 3. Aplikasi Menampilkan kalori yang dibutuhkan pengguna. |
| Alternative Flow | |

r. *Use case* Skenario Membalas Pesan

Pada *use case* Skenario Membalas Pesan akan dijelaskan lebih detail bagaimana proses seorang ahli gizi yang dikirim pesan oleh pengguna, kemudian dapat mengirim pesan balasan kepada pengguna tersebut Tabel 4.21 dibawah ini merupakan *use case* Skenario Membalas Pesan

Tabel 4.21 *Use case* Skenario Membalas Pesan

| | |
|----------------------|--|
| Nama <i>Use case</i> | Membalas Pesan |
| Aktor | Ahli gizi |
| Tujuan | Ahli gizi dapat membalas pesan yang dikirim oleh pengguna |
| Deskripsi | <i>Use case</i> ini berfungsi bagi ahli gizi untuk membalas pesan yang dikirim oleh pengguna. |
| Kondisi Awal | Aktor telah berhasil <i>login</i> dan masuk ke halaman utama. |
| Kondisi Akhir | Aplikasi berhasil membalas pesan kepada pengguna yang dituju |
| Main Flow | <ol style="list-style-type: none"> 1. Aktor memilih pesan yang akan dibalas 2. aktor memilih tombol balas yang ada disamping <i>List</i> pesan 3. aplikasi menampilkan <i>form</i> pesan yang digunakan aktor untuk menulis pesan 4. aktor menulis pesan balasan 5. aplikasi mengirim pesan kepada pengguna yang dituju sesuai dengan pengirimnya |
| Alternative Flow | |

s. *Use case* Skenario Melakukan *Logout* Ahli Gizi

Pada *use case* Skenario melakukan *logout* akan dijelaskan lebih detail bagaimana proses seorang pengguna melakukan *logout* terhadap aplikasi. Tabel 4.14 dibawah ini merupakan *use case* Skenario melakukan *logout* yang dilakukan oleh pengguna.

Tabel 4.22 *Use case* Skenario Melakukan *Logout* Ahli Gizi

| | |
|----------------------|---|
| Nama <i>Use case</i> | Melakukan <i>Logout</i> Ahli Gizi |
| Aktor | Pengguna |
| Tujuan | Pengguna dapat melakukan <i>logout</i> terhadap aplikasi |
| Deskripsi | <i>Use case</i> ini berfungsi bagi pengguna untuk melakukan <i>logout</i> . |

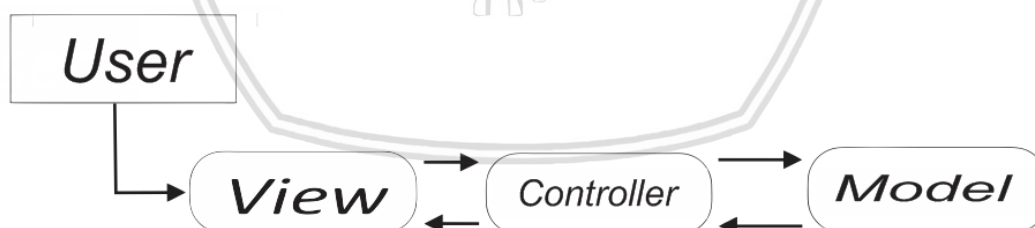
| | |
|------------------|--|
| Kondisi Awal | Aktor telah berhasil <i>login</i> dan masuk ke halaman utama |
| Kondisi Akhir | Aplikasi berhasil <i>logout</i> dari akun aktor |
| Main Flow | <ol style="list-style-type: none"> 4. Aktor memilih pilihan <i>logout</i> pada aplikasi 5. Aplikasi berhasil <i>logout</i> dari akun aktor 6. Aplikasi menampilkan halaman <i>login</i> |
| Alternative Flow | |

4.2 Perancangan

Tahapan yang dilakukan didalam perancangan sistem yaitu merancang arsitektural sistem, perancangan perhitungan kalori menggunakan metode *harris benedict*, perancangan penentuan menu makanan, pembuatan *sequence diagram*, *class diagram* dan perancangan antar muka.

4.2.1 Perancangan Arsitektural

Metode MVC merupakan metode yang dipilih sebagai perancangan arsitektural aplikasi ini. Dimana controller merupakan penghubung antara view dan model. *view* merupakan tampilan yang berhubungan langsung dengan pengguna sedangkan model adalah tempat data dan metode yang digunakan untuk melakukan perhitungan kalori menggunakan metode *harris benedict*. gambar 4.2 di gambarkan bagaimana aliran kerja dari metode MVC. lalu tabel 4.17 mendeskripsikan elemen - elemen dari MVC.



Gambar 4.2 Model View Controller

Tabel 4.23 Deskripsi Elemen MVC

| Nama | Deskripsi |
|-------------|--|
| <i>User</i> | Aktor yang melakukan aksi terhadap aplikasi nantinya |
| <i>View</i> | Merupakan unit tampilan yang berhubungan langsung dengan <i>user</i> |

| | |
|------------|---|
| Controller | Adalah unit penghubung antara <i>view</i> dan model |
| Model | Model merupakan tempat data dan metode yang digunakan untuk melakukan perhitungan |

4.2.2 Perancangan Perhitungan Kebutuhan Kalori Tubuh Menggunakan Metode *Harris benedict*

Untuk menghitung kalori khusus yang dibutuhkan oleh penyandang penyakit diabetes menggunakan metode *harris benedict*, diperlukan beberapa variabel pendukung untuk perhitungan kalori tersebut. Dimana rumus *harris benedict* adalah sebagai berikut

Perempuan : $65,5 + (9,6 \times \text{BB}) + (1,8 \times \text{TB}) - (4,7 \times \text{U})$

Laki-laki : $66 + (13,7 \times \text{BB}) + (5 \times \text{TB}) - (6,8 \times \text{U})$

Keterangan

BB : Berat badan dalam Kg

TB : Tinggi badan dalam Cm

U : Usia dalam Tahun

Contoh kasus seorang perempuan bernama rini terindikasi mengidap penyakit diabetes ia berumur 35 tahun memiliki tinggi badan 150cm dan berat badan 50 kg . Dari contoh kasus tersebut dilakukanlah perhitungan menggunakan rumus *harris benedict* sebagai berikut

Diketahui:

BB = 50 kg

TB = 150cm

U = 35

Jenis Kelamin = Perempuan

Lalu hasil tersebut dikalikan dengan faktor aktivitas dan faktor trauma atau stress yang sedang dialami oleh pasien pada table 4.23 berikut adalah tabel faktor aktivitas

Tabel 4.24 Tabel Faktor Aktivitas Dan Faktor Trauma

Sumber : (Almatsier, 2004)

| No | Aktifitas | Faktor | No | Jenis trauma / stres | Faktor |
|----|-------------------------------|--------|----|---|--------|
| 1. | Istirahat di tempat tidur | 1,2 | 1. | Tidak ada stres, pasien dalam keadaan gizi baik | 1,3 |
| 2. | Tidak terikat di tempat tidur | 1,3 | 2. | Stres ringan: peradangan saluran cerna, kanker, | 1,4 |

| | | | | |
|--|--|----|---|-----|
| | | | bedah elektif, trauma kerangka moderat | |
| | | 3. | Stres sedang : sepsis, bedah tulang, luka bakar, trauma kerangka mayor | 1,5 |
| | | 4. | Stres berat : trauma multiple, sepsis, dan bedah multisistem | 1,6 |
| | | 5. | Stres sangat berat : luka kepala berat, sindroma penyakit pernapasan akut, luka bakar dan sepsis Luka bakar sangat berat | 1,7 |

Angka Metabolisme Basal rini adalah

(rumus *harris benedict* bagi perempuan) :

$$AMB = 655 + (9,6 \times BB) + (1,8 \times TB) - (4,7 \times U)$$

$$AMB = 655 + (9,6 \times 50) + (1,8 \times 150) - (4,7 \times 35)$$

$$AMB = 655 + 480 + 270 - 164,5$$

$$AMB = 1240,5$$

Perhitungan Kalori yang dibutuhkan adalah :

$$\text{Energi} = \text{AMB} \times \text{Faktor Aktifitas}$$

$$\text{Energi} = 1240,5 \times 1,3 \times 1,3$$

$$= 2096,45 \text{ kkal}$$

$$= 2096 \text{ kkal (dibulatkan)}$$

4.2.3 Perancangan Penentuan Menu Makanan dengan Metode *Harris benedict*

Dalam penentuan menu makanan, sebelumnya harus dilakukan penentuan kalori yang dibutuhkan oleh tubuh. Sehingga kalori yang terdapat pada menu makanan mencukupi kebutuhan tubuh tidak melebihi dan tidak kurang. Kebutuhan kalori perhari setiap manusia berbeda – beda. Berdasarkan rekomendasi Kemenkes Ri dimana angka kecukupan gizi pada wanita dewasa yang berumur 16 – 30 tahun pada umumnya berkisar antara 2,125 sampai 2,250 kalori per hari. Sementara pria dewasa dengan rentang umur 16 – 30 tahun yaitu 2625-

2725 kalori per hari. Dibuatlah rancangan menu makanan dengan melakukan wawancara dengan *expert* di bidang ahli gizi.

4.2.3.1 Perancangan Menu Makanan Bagi Pria Dewasa

Perancangan menu makanan bagi pria dibagi menjadi 3 klasifikasi berdasarkan rekomendasi kebutuhan gizi dari Kemenkes RI yaitu pria dengan kebutuhan kalori 2725 kalori per hari, pria dengan kebutuhan kalori antara 2625-2725 kalori perhari dan pria dengan kebutuhan kalori 2625 kalori perhari. Pembagian menu makanan perharinya dibagi berdasarkan tabel 4.24 dibawah ini yang merupakan tabel persentase asupan makanan yang dianjurkan oleh para ahli

Tabel 4.25 Tabel persentase asupan makanan

(Almatsier, 2004)

| Waktu Makan | Pukul | Persentase |
|----------------|-------|---------------------------|
| Makan pagi | 07:00 | 20% × total kalori sehari |
| Selingan pagi | 10:00 | 10% × total kalori sehari |
| Makan siang | 12:00 | 30% × total kalori sehari |
| Selingan sore | 16:00 | 10% × total kalori sehari |
| Makan malam | 18:00 | 20% × total kalori sehari |
| Selingan malam | 21:00 | 10% × total kalori sehari |
| Total 100 % | | |

- a. Perancangan Menu Makanan Pria dengan Kebutuhan Kalori 2725 kalori perhari

Berdasarkan kebutuhan kalori pria dewasa sebanyak 2725 kalori perhari didapatkan sebuah pembagian persentase kalori porsi asupan menu makanan pria pada table 4.25 berikut

Tabel 4.26 Persentase Kalori Porsi Asupan Makanan Pria dengan Kebutuhan Kalori 2725 kalori perhari

| Waktu Makan | Pukul | Persentase |
|-------------------------|-------|---|
| Makan pagi | 07:00 | $20\% \times 2725 = 545 \text{ kkal}$ |
| Selingan pagi | 10:00 | $10\% \times 2725 = 272,5 \text{ kkal}$ |
| Makan siang | 12:00 | $30\% \times 2725 = 817,5 \text{ kkal}$ |
| Selingan sore | 16:00 | $10\% \times 2725 = 272,5 \text{ kkal}$ |
| Makan malam | 18:00 | $20\% \times 2725 = 545 \text{ kkal}$ |
| Selingan malam | 21:00 | $10\% \times 2725 = 272,5 \text{ kkal}$ |
| Total 100 % = 2725 kkal | | |

Dari persentase diatas didapatkan sebuah menu makanan yang cocok dan sesuai dengan kebutuhan kalori tersebut. Menu makanan didapat berdasarkan tabel kalori pada makanan yang sudah dibuat departemen kesehatan tahun 2011.

Tabel 4.26 menjelaskan rincian menu makan pagi bagi pria dewasa dengan kebutuhan kalori 2725 kalori perhari. Sesuai dengan pembagian porsi asupan menu makanan yang sudah dibuat pada tabel 4.25.

Tabel 4.27 Tabel Menu Makan Pagi Pria dengan Kebutuhan Kalori 2725 kalori perhari

| Nama Makanan | Berat (gr) | Kalori | Unit |
|---------------|-------------|--------------------|------|
| Bubur Tawar | 200 | 44 | 1 |
| Sop Bayam | 150 | 234 | 1 |
| Telur dadar | 40 | 100 | 0.75 |
| Apel Merah | 140 | 82 | 1 |
| Nama Minuman | Jumlah (ml) | Kalori | Unit |
| Teh (Cangkir) | 150 | 80 | 1 |
| Total (kkal) | | $(\pm) 545 (20\%)$ | |

Tabel 4.27 menjelaskan rincian menu selingan pagi bagi pria dewasa dengan kebutuhan kalori 2725 kalori perhari. Sesuai dengan pembagian porsi asupan menu makanan yang sudah dibuat pada tabel 4.25.

Tabel 4.28 Tabel Menu Selingan Pagi Pria dengan Kebutuhan Kalori 2725 kalori perhari

| Nama Makanan | Berat (gr) | Kalori | Unit |
|--------------|------------|----------------|------|
| Pisang | 125 | 270 | 2 |
| Total (kkal) | | (±)272,5 (10%) | |

Tabel 4.28 menjelaskan rincian menu makan siang bagi pria dewasa dengan kebutuhan kalori 2725 kalori perhari. Sesuai dengan pembagian porsi asupan menu makanan yang sudah dibuat pada tabel 4.25.

Tabel 4.29 Tabel Menu Makan Siang Pria dengan Kebutuhan Kalori 2725 kalori perhari

| Nama Makanan | Berat (gr) | Kalori | Unit |
|--------------------|-------------|----------------|------|
| Nasi Putih | 100 | 175 | 1 |
| Telur Ayam Rebus | 60 | 97 | 1 |
| Sop Mutiara Jagung | 200 | 126 | 1 |
| Apel | 320 | 184 | 2 |
| Pir | 440 | 200 | 2 |
| Nama Minuman | Jumlah (ml) | Kalori | Unit |
| Jus Melon | 150 | 35 | 1 |
| Total (kkal) | | (±)817,5 (30%) | |

Tabel 4.29 menjelaskan rincian menu selingan sore bagi pria dewasa dengan kebutuhan kalori 2725 kalori perhari. Sesuai dengan pembagian porsi asupan menu makanan yang sudah dibuat pada tabel 4.25.

Tabel 4.30 Tabel Menu Selingan Sore Pria dengan Kebutuhan Kalori 2725 kalori perhari

| Nama Makanan | Berat (gr) | Kalori | Unit |
|--------------|------------|----------------|------|
| Kue Ape | 210 | 270 | 3 |
| Total (kkal) | | (±)272,5 (10%) | |

Tabel 4.30 menjelaskan rincian menu makan malam bagi pria dewasa dengan kebutuhan kalori 2725 kalori perhari. Sesuai dengan pembagian porsi asupan menu makanan yang sudah dibuat pada tabel 4.25.

Tabel 4.31 Tabel Menu Makan Malam Pria dengan Kebutuhan Kalori 2725 kalori perhari

| Nama Makanan | Berat (gr) | Kalori | Unit |
|---------------|-------------|--------------|------|
| Nasi Putih | 100 | 175 | 1 |
| Ayam Rebus | 100 | 250 | 0.4 |
| Melon | 200 | 60 | 1 |
| Nama Minuman | Jumlah (ml) | Kalori | Unit |
| Teh (Cangkir) | 100 | 60 | 1 |
| Total (kkal) | | (±)545 (20%) | |

Tabel 4.31 menjelaskan rincian menu makan malam bagi pria dewasa dengan kebutuhan kalori 2725 kalori perhari. Sesuai dengan pembagian porsi asupan menu makanan yang sudah dibuat pada tabel 4.25.

Tabel 4.32 Tabel Menu Selingan Malam Pria dengan Kebutuhan Kalori 2725 kalori perhari

| Nama Makanan | Berat (gr) | Kalori | Unit |
|--------------|------------|----------------|------|
| susu | 100 | 42 | 1 |
| Roti Tawar | 100 | 220 | 2 |
| Total (kkal) | | (±)272,5 (10%) | |

- b. Perancangan Menu Makanan Pria dengan Kebutuhan Kalori antara 2625-2725 kalori perhari.

Berdasarkan kebutuhan kalori pria dewasa sebanyak 2625 -2725 kalori perhari didapatkan sebuah pembagian persentase kalori porsi asupan menu makanan pria pada table 4.32. Tabel tersebut didapat berdasarkan hasil diskusi dan pertimbangan yang telah dilakukan Bersama dengan *expert* ibu Ika dibidang ahli gizi.

Tabel 4.33 tabel pembagian porsi makan Pria dengan Kebutuhan Kalori antara 2625-2725 kalori perhari

| Waktu Makan | Pukul | Persentase |
|---------------|-------|-------------------------------|
| Makan pagi | 07:00 | $20\% \times 2675 = 535$ kkal |
| Selingan pagi | 10:00 | $10\% \times 2675 = 266$ kkal |
| Makan siang | 12:00 | $30\% \times 2675 = 800$ kkal |
| Selingan sore | 16:00 | $10\% \times 2675 = 266$ kkal |
| Makan malam | 18:00 | $20\% \times 2675 = 535$ kkal |

| | | |
|-------------------------|-------|-----------------------|
| Selingan malam | 21:00 | 10% × 2675 = 266 kkal |
| Total 100 % = 2675 kkal | | |

Dari persentase diatas didapatkan sebuah menu makanan yang cocok dan sesuai dengan kebutuhan kalori tersebut. Menu makanan didapat berdasarkan tabel yang sudah dibuat departemen kesehatan tahun 2011

Tabel 4.33 menjelaskan rincian menu makan pagi bagi pria dewasa dengan kebutuhan kalori antara 2625-2725 kalori perhari. Sesuai dengan pembagian porsi asupan menu makanan yang sudah dibuat pada tabel 4.32.

Tabel 4.34 Tabel Menu Makan Pagi Pria dengan Kebutuhan Kalori antara 2625-2725 kalori perhari

| Nama Makanan | Berat (gr) | Kalori | Unit |
|---------------|-------------|-------------|------|
| Jagung Rebus | 312 | 112 | 1,4 |
| Pisang Hijau | 200 | 105 | 1 |
| Telur dadar | 40 | 140 | 1 |
| Apel | 160 | 92 | 1 |
| Nama Minuman | Jumlah (ml) | Kalori | Unit |
| Teh (Cangkir) | 150 | 80 | 1 |
| Total (kkal) | | (±)535(20%) | |

Tabel 4.34 menjelaskan rincian menu selingan pagi bagi pria dewasa dengan kebutuhan kalori antara 2625-2725 kalori perhari. Sesuai dengan pembagian porsi asupan menu makanan yang sudah dibuat pada tabel 4.32.

Tabel 4.35 Tabel Menu Selingan Pagi Pria dengan Kebutuhan Kalori antara 2625-2725 kalori perhari

| Nama Makanan | Berat (gr) | Kalori | Unit |
|--------------|-------------|--------------|------|
| Pir | 440 | 200 | 2 |
| Nama Minuman | Jumlah (ml) | Kalori | Unit |
| Teh Tawar | 100 | 40 | 1 |
| Total (kkal) | | (±)266 (10%) | |

Tabel 4.35 menjelaskan rincian menu makan siang bagi pria dewasa dengan kebutuhan kalori antara 2625-2725 kalori perhari. Sesuai dengan pembagian porsi asupan menu makanan yang sudah dibuat pada tabel 4.32.

Tabel 4.36 Tabel Menu Makan Siang Pria dengan Kebutuhan Kalori antara 2625-2725 kalori perhari

| Nama Makanan | Berat (gr) | Kalori | Unit |
|---------------------|--------------------|---------------|-------------|
| Nasi Putih | 100 | 175 | 1 |
| Ikan Teri | 50 | 213 | 2,75 |
| Telur Ayam Rebus | 60 | 97 | 1 |
| Tumis Buncis | 100 | 52 | 1,5 |
| Pir | 440 | 200 | 2 |
| Jambu Air | 60 | 35 | 0,5 |
| Nama Minuman | Jumlah (ml) | Kalori | Unit |
| Jus Melon | 150 | 35 | 1 |
| Total (kkal) | | (±)800 (30%) | |

Tabel 4.36 menjelaskan rincian menu makan sore bagi pria dewasa dengan kebutuhan kalori antara 2625-2725. Sesuai dengan pembagian porsi asupan menu makanan yang sudah dibuat pada tabel 4.32.

Tabel 4.37 Tabel Menu Selingan Sore Pria dengan Kebutuhan Kalori antara 2625-2725 kalori perhari

| Nama Makanan | Berat (gr) | Kalori | Unit |
|---------------------|-------------------|---------------|-------------|
| Apel Merah | 140 | 41 | 0,5 |
| Pepaya | 100 | 46 | 0,5 |
| Jambu Biji | 320 | 157 | 2 |
| Total (kkal) | | (±)266 (10%) | |

Tabel 4.37 menjelaskan rincian menu makan malam bagi pria dewasa dengan kebutuhan kalori antara 2625-2725 kalori perhari. Sesuai dengan pembagian porsi asupan menu makanan yang sudah dibuat pada tabel 4.32.

Tabel 4.38 Tabel Menu Makan Malam Pria dengan Kebutuhan Kalori antara 2625-2725 kalori perhari

| Nama Makanan | Berat (gr) | Kalori | Unit |
|---------------------|--------------------|---------------|-------------|
| Nasi Putih | 100 | 175 | 1 |
| Ayam Rebus | 100 | 250 | 0.4 |
| Tumis Buncis | 200 | 70 | 2 |
| Nama Minuman | Jumlah (ml) | Kalori | Unit |
| Teh Tawar | 100 | 40 | 1 |
| Total (kkal) | | (±)535 (20%) | |

Tabel 4.38 menjelaskan rincian menu selingan malam bagi pria dewasa dengan kebutuhan kalori antara 2625-2725 kalori perhari. Sesuai dengan pembagian porsi asupan menu makanan yang sudah dibuat pada tabel 4.32.

Tabel 4.39 Tabel Menu Selingan Malam Pria dengan Kebutuhan Kalori antara 2625-2725 kalori perhari

| Nama Makanan | Berat (gr) | Kalori | Unit |
|--------------|------------|--------------|------|
| susu | 100 | 42 | 1 |
| Roti Tawar | 50 | 110 | 1 |
| Pepaya | 200 | 132 | 1 |
| Total (kkal) | | (±)266 (10%) | |

- c. Perancangan Menu Makanan Pria dengan Kebutuhan Kalori 2625 Kalori perhari.

Berdasarkan kebutuhan kalori pria dewasa sebanyak 2625 kalori perhari didapatkan sebuah pembagian persentase kalori porsi asupan menu makanan pria pada table 4.39. Tabel tersebut didapat berdasarkan hasil diskusi dan pertimbangan yang telah dilakukan Bersama dengan *expert* ibu Ika dibidang ahli gizi.

Tabel 4.40 Tabel Pembagian Porsi Makan Pria dengan Kebutuhan Kebutuhan Kalori 2625 Kalori perhari.

| Waktu Makan | Pukul | Persentase |
|-------------------------|-------|-------------------------------|
| Makan pagi | 07:00 | $20\% \times 2625 = 525$ kkal |
| Selingan pagi | 10:00 | $10\% \times 2625 = 262$ kkal |
| Makan siang | 12:00 | $30\% \times 2625 = 786$ kkal |
| Selingan sore | 16:00 | $10\% \times 2625 = 268$ kkal |
| Makan malam | 18:00 | $20\% \times 2625 = 525$ kkal |
| Selingan malam | 21:00 | $10\% \times 2625 = 262$ kkal |
| Total 100 % = 2625 kkal | | |

Dari persentase diatas didapatlah sebuah menu makanan yang cocok dan sesuai dengan kebutuhan kalori tersebut. Menu makanan didapat berdasarkan tabel yang sudah dibuat departemen kesehatan tahun 2011.

Tabel 4.40 menjelaskan rincian menu makan pagi bagi pria dewasa dengan kebutuhan kalori 2625 kalori perhari. Sesuai dengan pembagian porsi asupan menu makanan yang sudah dibuat pada tabel 4.39.

Tabel 4.41 Tabel Menu Makan Pagi Pria dengan Kebutuhan Kebutuhan Kalori 2625 Kalori perhari.

| Nama Makanan | Berat (gr) | Kalori | Unit |
|--------------|-------------|-------------|------|
| Bubur | 300 | 88 | 1 |
| Telur dadar | 80 | 280 | 2 |
| Apel | 160 | 92 | 1 |
| Nama Minuman | Jumlah (ml) | Kalori | Unit |
| Teh Tawar | 100 | 40 | 1 |
| Total (kkal) | | (±)525(20%) | |

Tabel 4.41 menjelaskan rincian menu selingan pagi bagi pria dewasa dengan kebutuhan kalori 2625 kalori perhari. Sesuai dengan pembagian porsi asupan menu makanan yang sudah dibuat pada tabel 4.39.

Tabel 4.42 Tabel Menu Selingan Pagi Pria dengan Kebutuhan Kalori 2625 kalori perhari

| Nama Makanan | Berat (gr) | Kalori | Unit |
|--------------|-------------|--------------|------|
| Crakers | 50 | 229 | 2,75 |
| Nama Minuman | Jumlah (ml) | Kalori | Unit |
| Teh Tawar | 100 | 40 | 1 |
| Total (kkal) | | (±)262 (10%) | |

Tabel 4.42 menjelaskan rincian menu makan siang bagi pria dewasa dengan kebutuhan kalori 2625 kalori perhari. Sesuai dengan pembagian porsi asupan menu makanan yang sudah dibuat pada tabel 4.39.

Tabel 4.43 Tabel Menu Makan Siang Pria dengan Kebutuhan Kalori 2625 Kalori Perhari

| Nama Makanan | Berat (gr) | Kalori | Unit |
|------------------|------------|--------|------|
| Nasi Putih | 100 | 175 | 1 |
| Ikan Teri | 50 | 213 | 2,75 |
| Telur Ayam Rebus | 60 | 97 | 1 |
| Sayur Lodeh | 100 | 61 | 0,75 |
| Pir | 440 | 200 | 2 |

| Nama Minuman | Jumlah (ml) | Kalori | Unit |
|--------------|-------------|--------------|------|
| Jus Melon | 150 | 35 | 1 |
| Total (kkal) | | (±)786 (30%) | |

Tabel 4.43 menjelaskan rincian menu selingan sore bagi pria dewasa dengan kebutuhan kalori 2625 kalori perhari. Sesuai dengan pembagian porsi asupan menu makanan yang sudah dibuat pada tabel 4.39.

Tabel 4.44 Tabel Menu Selingan Sore Pria dengan Kebutuhan Kalori antara 2625 kalori perhari

| Nama Makanan | Berat (gr) | Kalori | Unit |
|--------------|------------|--------------|------|
| Apel Merah | 70 | 41 | 0,5 |
| Jambu Biji | 320 | 157 | 2 |
| Total (kkal) | | (±)262 (10%) | |

Tabel 4.44 menjelaskan rincian menu makan malam bagi pria dewasa dengan kebutuhan kalori 2625 kalori perhari. Sesuai dengan pembagian porsi asupan menu makanan yang sudah dibuat pada tabel 4.39.

Tabel 4.45 Tabel Menu Makan Malam Pria dengan Kebutuhan Kalori antara 2625-2725 kalori perhari

| Nama Makanan | Berat (gr) | Kalori | Unit |
|---------------------|-------------|--------------|------|
| Nasi Putih | 100 | 175 | 1 |
| Ayam Rebus | 100 | 250 | 0.4 |
| Tumis Buncis | 100 | 35 | 1 |
| Nama Minuman | Jumlah (ml) | Kalori | Unit |
| Teh Tawar (Cangkir) | 100 | 80 | 1 |
| Total (kkal) | | (±)525 (20%) | |

Tabel 4.45 menjelaskan rincian menu makan malam bagi pria dewasa dengan kebutuhan kalori 2625 kalori perhari. Sesuai dengan pembagian porsi asupan menu makanan yang sudah dibuat pada tabel 4.39.

Tabel 4.46 Tabel Menu Selingan Malam Pria dengan Kebutuhan Kalori antara 2625-2725 kalori perhari

| Nama Makanan | Berat (gr) | Kalori | Unit |
|--------------|------------|--------------|------|
| susu | 100 | 42 | 1 |
| Roti Tawar | 50 | 110 | 1 |
| Pepaya | 200 | 132 | 1 |
| Total (kkal) | | (±)262 (10%) | |

4.2.3.2 Perancangan Menu Makanan Bagi Wanita Dewasa

Perancangan menu makanan bagi wanita dewasa dibagi menjadi 3 klasifikasi berdasarkan rekomendasi kebutuhan gizi dari Kemenkes RI yaitu wanita dengan kebutuhan kalori 2250 kalori per hari, wanita dengan kebutuhan kalori antara 2125-2250 kalori perhari dan wanita dengan kebutuhan kalori 2250 kalori perhari.

- a. Perancangan Menu Makanan Wanita dengan Kebutuhan Kalori 2250 Kalori perhari.

Berdasarkan kebutuhan kalori wanita dewasa sebanyak 2225 kalori perhari didapatkan sebuah pembagian persentase kalori porsi asupan menu makanan pria pada tabel 4.46. Tabel tersebut didapat berdasarkan hasil diskusi dan pertimbangan yang telah dilakukan Bersama dengan *expert* ibu Ika dibidang ahli gizi.

Tabel 4.47 Tabel Pembagian Porsi Makan wanita dengan Kebutuhan Kalori 2250 Kalori perhari.

| Waktu Makan | Pukul | Persentase |
|-------------------------|-------|---------------------------------------|
| Makan pagi | 07:00 | $20\% \times 2250 = 450 \text{ kkal}$ |
| Selingan pagi | 10:00 | $10\% \times 2250 = 225 \text{ kkal}$ |
| Makan siang | 12:00 | $30\% \times 2250 = 675 \text{ kkal}$ |
| Selingan sore | 16:00 | $10\% \times 2250 = 225 \text{ kkal}$ |
| Makan malam | 18:00 | $20\% \times 2250 = 450 \text{ kkal}$ |
| Selingan malam | 21:00 | $10\% \times 2250 = 225 \text{ kkal}$ |
| Total 100 % = 2250 kkal | | |

Dari persentase diatas didapatlah sebuah menu makanan yang cocok dan sesuai dengan kebutuhan kalori tersebut. Menu makanan didapat berdasarkan tabel yang sudah dibuat departemen kesehatan tahun 2011.

Tabel 4.47 menjelaskan rincian menu makan pagi bagi wanitadewasa dengan kebutuhan kalori 2250 kalori perhari. Sesuai dengan pembagian porsi asupan menu makanan yang sudah dibuat pada tabel 4.46

Tabel 4.48 Tabel Menu Makan Pagi Wanita Dengan Kebutuhan Kalori 2250 Kalori perhari.

| Nama Makanan | Berat (gr) | Kalori | Unit |
|--------------|-------------|-------------|------|
| Bubur Ayam | 200 | 165 | 1 |
| Telur dadar | 80 | 280 | 2 |
| Nama Minuman | Jumlah (ml) | Kalori | Unit |
| Teh Tawar | 100 | 40 | 1 |
| Total (kkal) | | (±)450(20%) | |

Tabel 4.48 menjelaskan rincian menu selingan pagi bagi wanita dewasa dengan kebutuhan kalori 2250 kalori perhari. Sesuai dengan pembagian porsi asupan menu makanan yang sudah dibuat pada tabel 4.46.

Tabel 4.49 Tabel Menu Selingan Pagi Wanita dengan Kebutuhan Kalori 2250 kalori perhari

| Nama Makanan | Berat (gr) | Kalori | Unit |
|--------------|------------|--------------|------|
| Crakers | 50 | 229 | 2,75 |
| Total (kkal) | | (±)225 (10%) | |

Tabel 4.49 menjelaskan rincian menu makan siang bagi pria dewasa dengan kebutuhan kalori 2250 kalori perhari. Sesuai dengan pembagian porsi asupan menu makanan yang sudah dibuat pada tabel 4.46.

Tabel 4.50 Tabel Menu Makan Siang Wanita Dengan Kebutuhan Kalori 2250 Kalori Perhari

| Nama Makanan | Berat (gr) | Kalori | Unit |
|------------------|-------------|--------------|------|
| Nasi Putih | 100 | 175 | 1 |
| Telur Ayam Rebus | 100 | 120 | 1,5 |
| Sayur Lodeh | 100 | 100 | 0,75 |
| Pir | 440 | 200 | 2 |
| Nama Minuman | Jumlah (ml) | Kalori | Unit |
| Jus Melon | 150 | 35 | 1 |
| Total (kkal) | | (±)675 (30%) | |

Tabel 4.50 menjelaskan rincian menu selingan sore bagi pria dewasa dengan kebutuhan kalori 2250 kalori perhari. Sesuai dengan pembagian porsi asupan menu makanan yang sudah dibuat pada tabel 4.49.

Tabel 4.51 Tabel Menu Selingan Sore Wanita Dengan Kebutuhan Kalori antara 2250 kalori perhari

| Nama Makanan | Berat (gr) | Kalori | Unit |
|--------------|------------|--------------|------|
| Crakers | 50 | 229 | 2,75 |
| Total (kkal) | | (±)225 (10%) | |

Tabel 4.51 menjelaskan rincian menu makan malam bagi pria dewasa dengan kebutuhan kalori 2225 kalori perhari. Sesuai dengan pembagian porsi asupan menu makanan yang sudah dibuat pada tabel 4.46.

Tabel 4.52 Tabel Menu Makan Malam Wanita Dengan Kebutuhan Kalori antara 2250 kalori perhari

| Nama Makanan | Berat (gr) | Kalori | Unit |
|--------------|-------------|--------------|------|
| Nasi Putih | 100 | 175 | 1 |
| Ayam Rebus | 100 | 250 | 0.4 |
| Tumis Buncis | 50 | 20 | 0.5 |
| Nama Minuman | Jumlah (ml) | Kalori | Unit |
| Teh Tawar | 100 | 40 | 1 |
| Total (kkal) | | (±)450 (20%) | |

Tabel 4.52 menjelaskan rincian menu makan malam bagi pria dewasa dengan kebutuhan kalori 2250 kalori perhari. Sesuai dengan pembagian porsi asupan menu makanan yang sudah dibuat pada tabel 4.46.

Tabel 4.53 Tabel Menu Selingan Malam Wanita dengan Kebutuhan Kalori antara 2250 Kalori Perhari

| Nama Makanan | Berat (gr) | Kalori | Unit |
|--------------|------------|--------------|------|
| Roti Tawar | 50 | 110 | 1 |
| Pepaya | 200 | 132 | 1 |
| Total (kkal) | | (±)262 (10%) | |

- b. Perancangan Menu Makanan Wanita dengan Kebutuhan Kalori Antara 2125-2550 Kalori per hari.

Berdasarkan kebutuhan kalori wanita dewasa antara 2125-2550 Kalori per hari. Didapatkan sebuah pembagian persentase kalori porsi asupan menu makanan pria pada tabel 4.53. Tabel tersebut didapat berdasarkan hasil diskusi dan pertimbangan yang telah dilakukan Bersama dengan *expert* ibu Ika dibidang ahli gizi.

Tabel 4.54 Tabel Pembagian Porsi Makan Wanita dengan Kebutuhan Kalori Antara 2125-2550 Kalori perhari.

| Waktu Makan | Pukul | Persentase |
|-------------------------|-------|---------------------------------------|
| Makan pagi | 07:00 | $20\% \times 2187 = 437 \text{ kkal}$ |
| Selingan pagi | 10:00 | $10\% \times 2187 = 218 \text{ kkal}$ |
| Makan siang | 12:00 | $30\% \times 2187 = 656 \text{ kkal}$ |
| Selingan sore | 16:00 | $10\% \times 2187 = 218 \text{ kkal}$ |
| Makan malam | 18:00 | $20\% \times 2187 = 437 \text{ kkal}$ |
| Selingan malam | 21:00 | $10\% \times 2187 = 218 \text{ kkal}$ |
| Total 100 % = 2187 kkal | | |

Dari persentase diatas didapatkanlah sebuah menu makanan yang cocok dan sesuai dengan kebutuhan kalori tersebut. Menu makanan didapat berdasarkan tabel yang sudah dibuat departemen kesehatan tahun 2011.

Tabel 4.54 menjelaskan rincian menu makan pagi bagi wanitadewasa dengan kebutuhan kalori antara 2125 - 2250 kalori perhari. Sesuai dengan pembagian porsi asupan menu makanan yang sudah dibuat pada tabel 4.53.

Tabel 4.55 Tabel Menu Makan Pagi Wanita Dengan Kebutuhan Kebutuhan Kalori Antara 2125-2550 Kalori perhari.

| Nama Makanan | Berat (gr) | Kalori | Unit |
|--------------|-------------|------------------|------|
| Bubur Ayam | 200 | 165 | 1 |
| Roti Tawar | 50 | 128 | 1,5 |
| Apel | 160 | 92 | 1 |
| Nama Minuman | Jumlah (ml) | Kalori | Unit |
| Jus Melon | 170 | 50 | 0,7 |
| Total (kkal) | | $(\pm)437(20\%)$ | |

Tabel 4.55 menjelaskan rincian menu selingan pagi bagi wanita dewasa dengan kebutuhan kalori antara 2125-2550 kalori perhari. Sesuai dengan pembagian porsi asupan menu makanan yang sudah dibuat pada tabel 4.53.

Tabel 4.56 Tabel Menu Selingan Pagi Wanita dengan Kebutuhan Kalori Antara 2125-2550 kalori perhari

| Nama Makanan | Berat (gr) | Kalori | Unit |
|--------------|------------|--------------|------|
| Pisang Rebus | 125 | 136,5 | 1,75 |
| Apel | 160 | 92 | 1 |
| Total (kkal) | | (±)218 (10%) | |

Tabel 4.56 menjelaskan rincian menu makan siang bagi pria dewasa dengan kebutuhan kalori antara 2125-2550 kalori perhari. Sesuai dengan pembagian porsi asupan menu makanan yang sudah dibuat pada tabel 4.53.

Tabel 4.57 Tabel Menu Makan Siang Wanita Dengan Kebutuhan Kalori Antara 2125-2550 Kalori Perhari

| Nama Makanan | Berat (gr) | Kalori | Unit |
|-----------------|-------------|--------------|------|
| Nasi Putih | 100 | 175 | 1 |
| Sayur Asam | 120 | 115 | 1,5 |
| Ikan Teri | 50 | 213 | 2,75 |
| Perkedel Jagung | 50 | 108 | 1,25 |
| Nama Minuman | Jumlah (ml) | Kalori | Unit |
| Jus Melon | 150 | 35 | 1 |
| Total (kkal) | | (±)656 (30%) | |

Tabel 4.57 menjelaskan rincian menu selingan sore bagi pria dewasa dengan kebutuhan kalori antara 2125-2550 kalori perhari. Sesuai dengan pembagian porsi asupan menu makanan yang sudah dibuat pada tabel 4.53.

Tabel 4.58 Tabel Menu Selingan Sore Wanita Dengan Kebutuhan Kalori antara 2125 - 2250 kalori perhari

| Nama Makanan | Berat (gr) | Kalori | Unit |
|--------------|------------|--------------|------|
| Crakers | 50 | 229 | 2,75 |
| Total (kkal) | | (±)218 (10%) | |

Tabel 4.58 menjelaskan rincian menu makan malam bagi pria dewasa dengan kebutuhan kalori 2125-2550 kalori perhari. Sesuai dengan pembagian porsi asupan menu makanan yang sudah dibuat pada tabel 4.53.

Tabel 4.59 Tabel Menu Makan Malam Wanita Dengan Kebutuhan Kalori antara 2125 - 2250 kalori perhari

| Nama Makanan | Berat (gr) | Kalori | Unit |
|--------------|-------------|--------|------|
| Nasi Putih | 70 | 120 | 0,7 |
| Ayam Rebus | 100 | 250 | 0.4 |
| Nama Minuman | Jumlah (ml) | Kalori | Unit |

| | | | |
|--------------|-----|--------------|---|
| Teh Tawar | 100 | 40 | 1 |
| Total (kkal) | | (±)437 (20%) | |

Tabel 4.59 menjelaskan rincian menu makan malam bagi pria dewasa dengan kebutuhan kalori 2125 - 2250 kalori perhari. Sesuai dengan pembagian porsi asupan menu makanan yang sudah dibuat pada tabel 4.53.

Tabel 4.60 Tabel Menu Selingan Malam Wanita Dengan Kebutuhan Kalori antara 2125 - 2250 Kalori Perhari

| Nama Makanan | Berat (gr) | Kalori | Unit |
|--------------|------------|--------------|------|
| Crakers | 50 | 229 | 2,75 |
| Total (kkal) | | (±)218 (10%) | |

- c. Perancangan Menu Makanan Wanita Dengan Kebutuhan Kalori 2125 Kalori per hari.

Berdasarkan kebutuhan kalori wanita dewasa 2125 Kalori per hari. Didapatkan sebuah pembagian persentase kalori porsi asupan menu makanan wanita pada tabel 4.60. Tabel tersebut didapat berdasarkan hasil diskusi dan pertimbangan yang telah dilakukan Bersama dengan *expert* ibu Ika dibidang ahli gizi.

Tabel 4.61 Tabel Pembagian Porsi Makan Wanita dengan Kebutuhan Kalori 2125 Kalori perhari.

| Waktu Makan | Pukul | Persentase |
|-------------------------|-------|-------------------------------|
| Makan pagi | 07:00 | $20\% \times 2125 = 425$ kkal |
| Selingan pagi | 10:00 | $10\% \times 2125 = 212$ kkal |
| Makan siang | 12:00 | $30\% \times 2125 = 637$ kkal |
| Selingan sore | 16:00 | $10\% \times 2125 = 212$ kkal |
| Makan malam | 18:00 | $20\% \times 2125 = 425$ kkal |
| Selingan malam | 21:00 | $10\% \times 2125 = 212$ kkal |
| Total 100 % = 2125 kkal | | |

Dari persentase diatas didapatlah sebuah menu makanan yang cocok dan sesuai dengan kebutuhan kalori tersebut. Menu makanan didapat berdasarkan tabel yang sudah dibuat departemen kesehatan tahun 2011.

Tabel 4.61 menjelaskan rincian menu makan pagi bagi wanita dewasa dengan kebutuhan kalori 2125 kalori perhari. Sesuai dengan pembagian porsi asupan menu makanan yang sudah dibuat pada tabel 4.60.

Tabel 4.62 Tabel Menu Makan Pagi Wanita Dengan Kebutuhan Kebutuhan Kalori 2125 Kalori perhari.

| Nama Makanan | Berat (gr) | Kalori | Unit |
|--------------|-------------|-------------|------|
| Telur dadar | 75 | 188 | 2,25 |
| Roti Tawar | 50 | 128 | 1,5 |
| Apel | 160 | 92 | 1 |
| Nama Minuman | Jumlah (ml) | Kalori | Unit |
| Jus Melon | 170 | 50 | 0,7 |
| Total (kkal) | | (±)425(20%) | |

Tabel 4.62 menjelaskan rincian menu selingan pagi bagi wanita dewasa dengan kebutuhan kalori 2125 kalori perhari. Sesuai dengan pembagian porsi asupan menu makanan yang sudah dibuat pada tabel 4.60.

Tabel 4.63 Tabel Menu Selingan Pagi Wanita dengan Kebutuhan Kalori 2125 kalori perhari

| Nama Makanan | Berat (gr) | Kalori | Unit |
|--------------|------------|--------------|------|
| Pisang Rebus | 125 | 136,5 | 1,75 |
| Salak | 150 | 63,6 | 0,75 |
| Total (kkal) | | (±)212 (10%) | |

Tabel 4.63 menjelaskan rincian menu makan siang bagi wanita dewasa dengan kebutuhan kalori 2125 kalori perhari. Sesuai dengan pembagian porsi asupan menu makanan yang sudah dibuat pada tabel 4.60.

Tabel 4.64 Tabel Menu Makan Siang Wanita Dengan Kebutuhan Kalori Antara 2125 Kalori Perhari

| Nama Makanan | Berat (gr) | Kalori | Unit |
|-----------------|-------------|--------|------|
| Nasi Putih | 100 | 175 | 2,25 |
| Ikan Teri | 50 | 213 | 2,75 |
| Perkedel Jagung | 50 | 108 | 1,25 |
| Sop Bayam | 50 | 78 | 1 |
| Jambu Air | 60 | 35,4 | 0,5 |
| Nama Minuman | Jumlah (ml) | Kalori | Unit |
| Jus Melon | 150 | 35 | 1 |

| | |
|--------------|--------------|
| Total (kkal) | (±)637 (30%) |
|--------------|--------------|

Tabel 4.64 menjelaskan rincian menu selingan sore bagi wanita dewasa dengan kebutuhan kalori antara 2125 kalori perhari. Sesuai dengan pembagian porsi asupan menu makanan yang sudah dibuat pada tabel 4.60.

Tabel 4.65 Tabel Menu Selingan Sore Wanita Dengan Kebutuhan Kalori 2125 kalori perhari

| Nama Makanan | Berat (gr) | Kalori | Unit |
|-----------------|--------------|--------|------|
| Jambu Biji | 320 | 157 | 2 |
| Jeruk Pontianak | 145 | 60 | 0,70 |
| Total (kkal) | (±)212 (10%) | | |

Tabel 4.65 menjelaskan rincian menu makan malam bagi wanita dewasa dengan kebutuhan kalori 2125 kalori perhari. Sesuai dengan pembagian porsi asupan menu makanan yang sudah dibuat pada tabel 4.60.

Tabel 4.66 Tabel Menu Makan Malam Wanita Dengan Kebutuhan Kalori antara 2125 - 2250 kalori perhari

| Nama Makanan | Berat (gr) | Kalori | Unit |
|--------------|--------------|--------|------|
| Nasi Putih | 70 | 120 | 0,7 |
| Ayam Rebus | 100 | 250 | 0.4 |
| Tumis Buncis | 100 | 52 | 1,5 |
| Total (kkal) | (±)425 (20%) | | |

Tabel 4.66 menjelaskan rincian menu makan malam bagi wanita dewasa dengan kebutuhan kalori 2125 kalori perhari. Sesuai dengan pembagian porsi asupan menu makanan yang sudah dibuat pada tabel 4.60.

Tabel 4.67 Tabel Menu Selingan Malam Wanita Dengan Kebutuhan Kalori antara 2125 Kalori Perhari

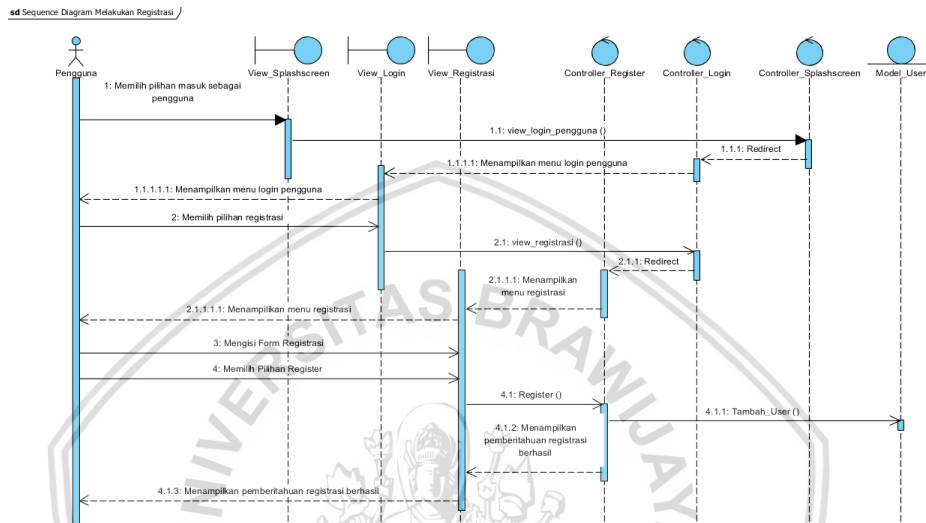
| Nama Makanan | Berat (gr) | Kalori | Unit |
|--------------|--------------|--------|------|
| Crakers | 45 | 210 | 2 |
| Total (kkal) | (±)218 (10%) | | |

4.3 Sequence Diagram

Pada *sequence diagram* akan dijelaskan bagaimana detail alur proses yang terjadi pada setiap *use case* yang ada didalam aplikasi ini.

4.3.1 Sequence Diagram Melakukan Registrasi

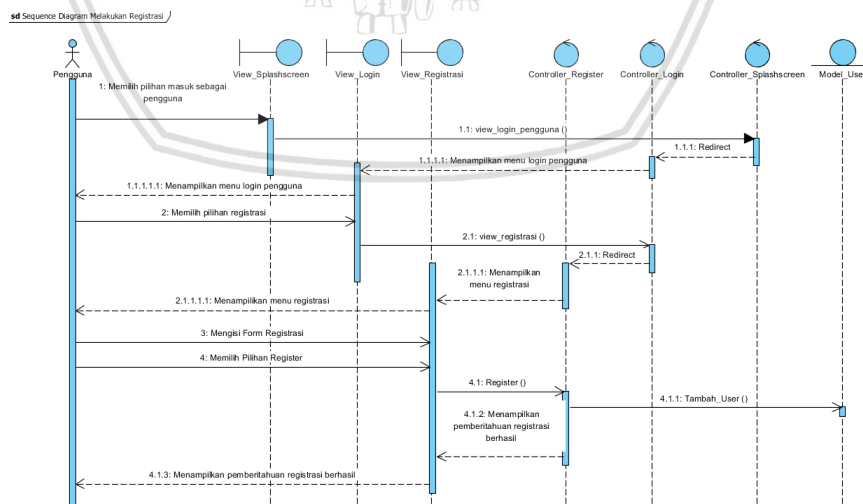
Gambar 4.3 dibawah ini merupakan deskripsi dari *sequence diagram* dari *use case* melakukan registrasi yang dilakukan oleh pengguna.



Gambar 4.3 Sequence Diagram Melakukan Registrasi

4.3.2 Sequence Diagram Melakukan Login

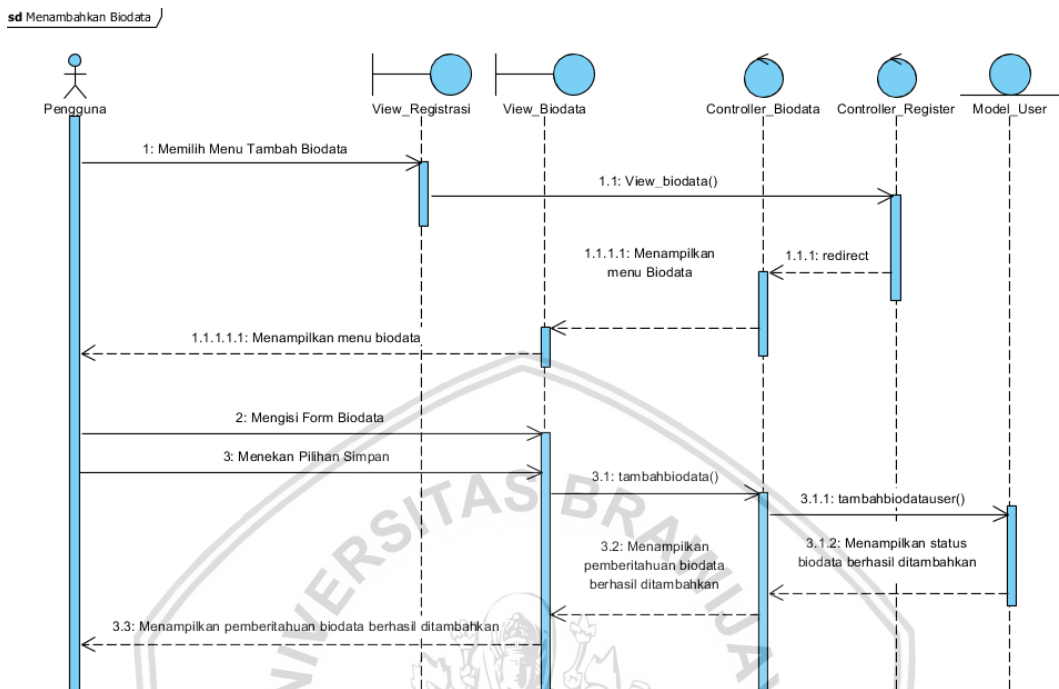
Gambar 4.4 dibawah ini merupakan deskripsi dari *sequence diagram* dari *use case* Melakukan Login dari aktor pengguna.



Gambar 4.4 Sequence Diagram Melakukan Login

4.3.3 Sequence Diagram Menambahkan Biodata

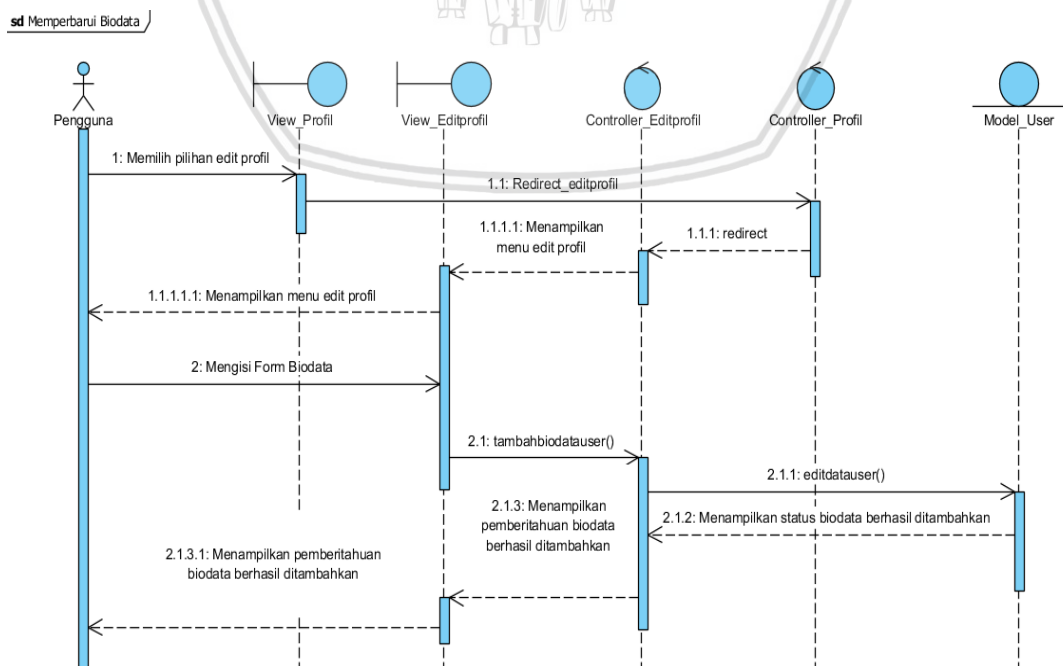
Gambar 4.5 dibawah ini merupakan deskripsi dari *sequence diagram* dari *use case* menambahkan biodata yang dilakukan oleh pengguna.



Gambar 4.5 Sequence Diagram Menambahkan Biodata

4.3.4 Sequence Diagram Memperbarui Biodata

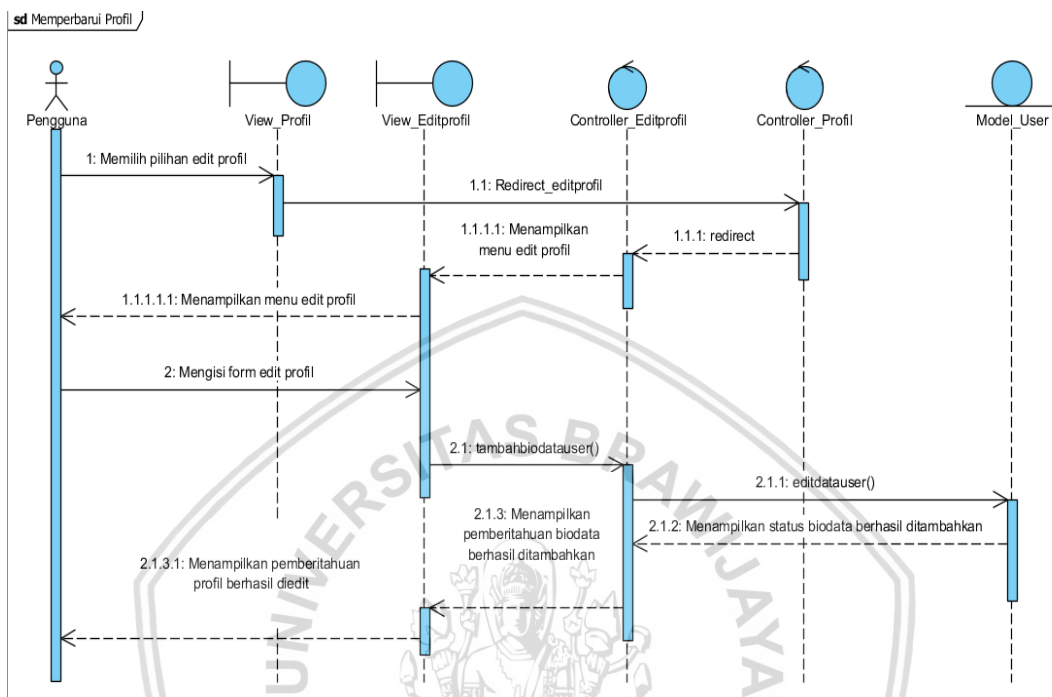
Gambar 4.6 dibawah ini merupakan deskripsi dari *sequence diagram* dari *use case* memperbarui biodata yang dilakukan oleh pengguna.



Gambar 4.6 Sequence Diagram Memperbarui Biodata

4.3.5 Sequence Diagram Memperbarui Profil

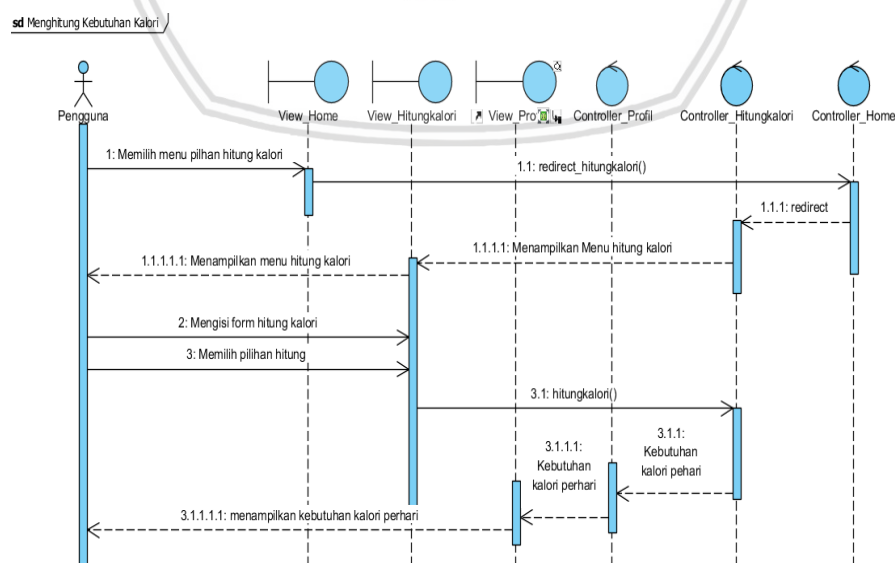
Gambar 4.7 dibawah ini merupakan deskripsi dari *sequence diagram* dari *use case* memperbarui profil yang dilakukan oleh pengguna.



Gambar 4.7 Sequence Diagram Memperbarui Profil

4.3.6 Sequence Diagram Menghitung Kebutuhan Kalori Perhari

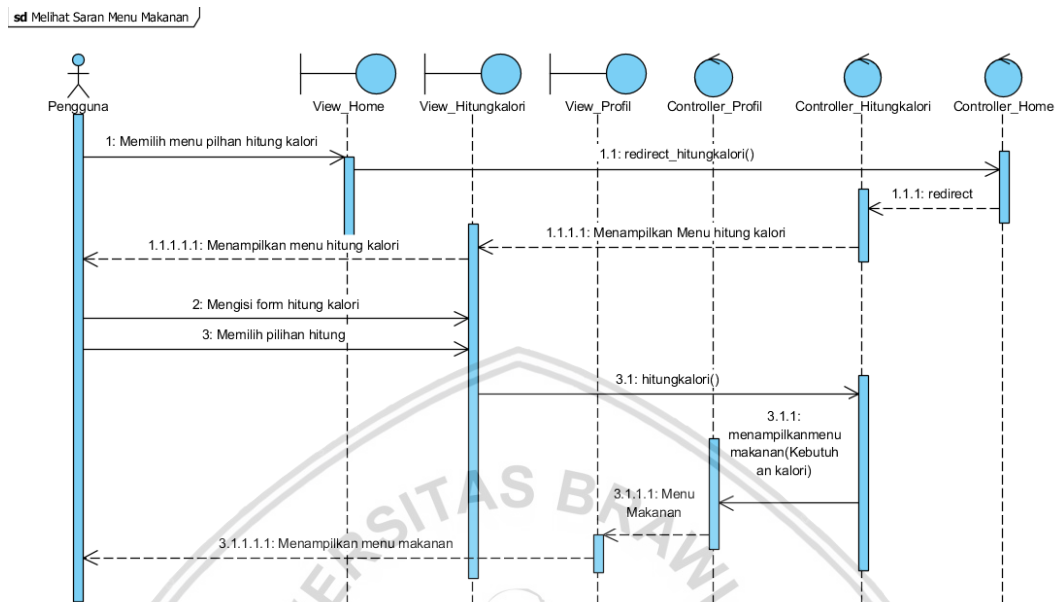
Gambar 4.8 dibawah ini merupakan deskripsi dari *sequence diagram* dari *use case* Menghitung Kalori yang dilakukan oleh pengguna.



Gambar 4.8 Sequence Diagram Menghitung Kalori Perhari

4.3.7 Sequence Diagram Melihat Saran Menu Makanan

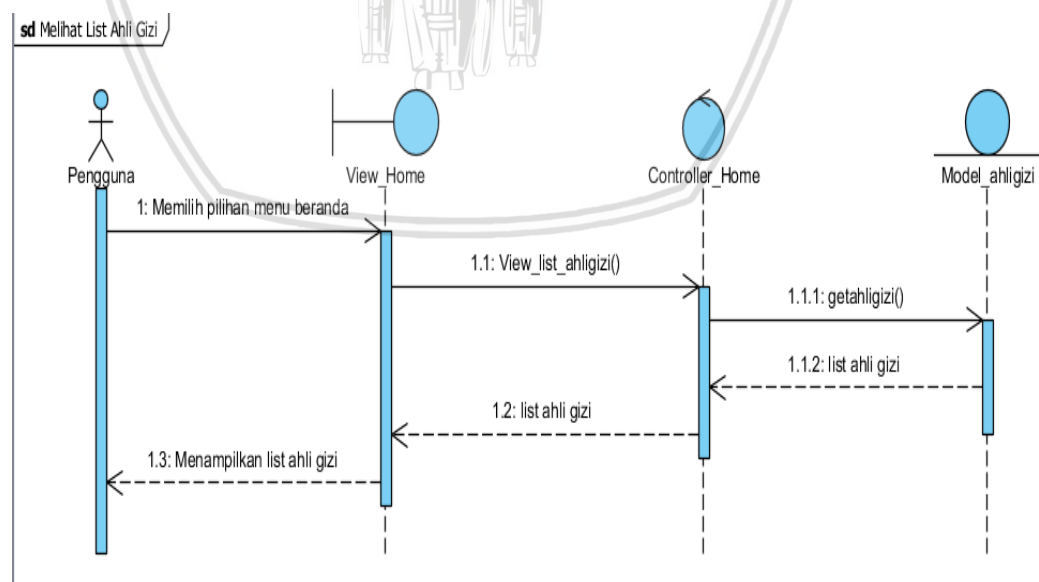
Gambar 4.9 dibawah ini merupakan deskripsi dari *sequence diagram* dari *use case* MelihatSaran Menu Makanan yang dilakukan oleh pengguna.



Gambar 4.9 Sequence Diagram Melihat Saran Menu Makanan

4.3.8 Sequence Diagram Melihat List Ahli Gizi

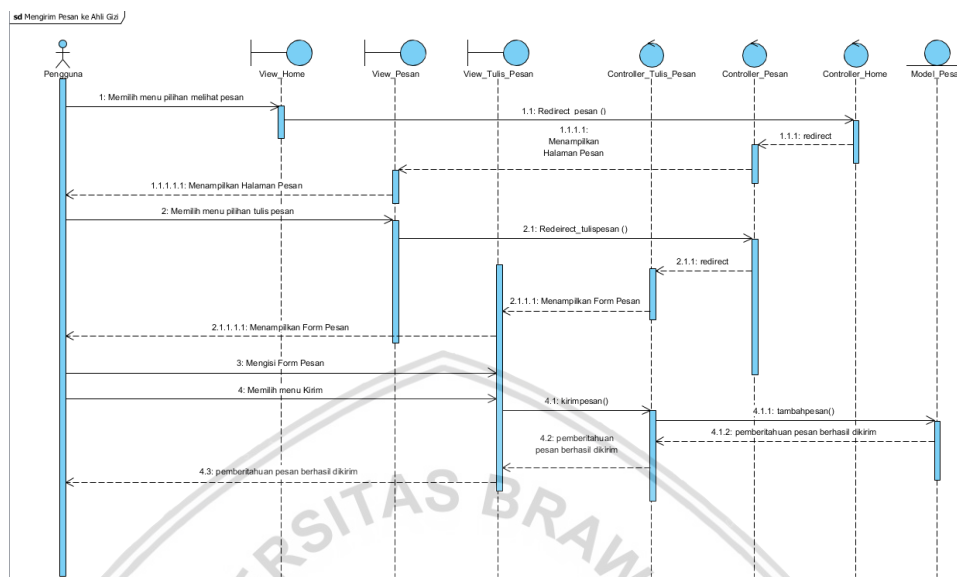
Gambar 4.10 dibawah ini merupakan deskripsi *sequence diagram* dari *use case* Melihat List Ahli Gizi.



Gambar 4.10 Sequence Diagram Melihat List Ahli Gizi

4.3.9 Sequence Diagram Mengirim Pesan ke Ahli Gizi

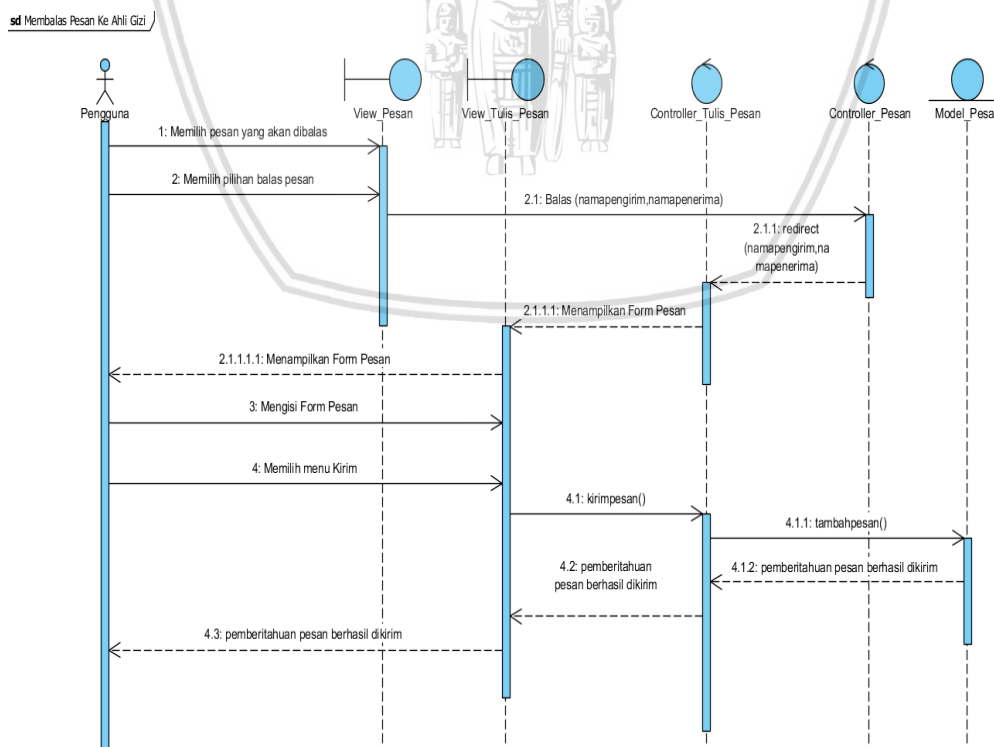
Gambar 4.11 dibawah ini merupakan deskripsi *sequence diagram* dari *use case* Mengirim Pesan ke Ahli Gizi yang dilakukan oleh pengguna.



Gambar 4.11 Sequence Diagram Mengirim Pesan ke Ahli Gizi

4.3.10 Sequence Diagram Membalas Pesan ke Ahli Gizi

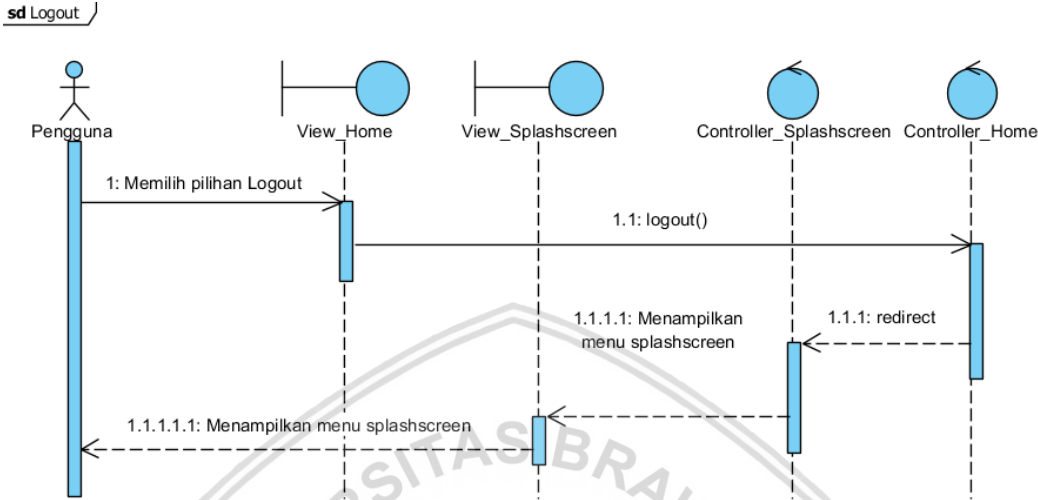
Gambar 4.12 dibawah ini merupakan deskripsi *sequence diagram* dari *use case* Membalas Pesan ke Ahli Gizi yang dilakukan oleh pengguna.



Gambar 4.12 Sequence Diagram Membalas Pesan

4.3.11 Sequence Diagram Melakukan Logout

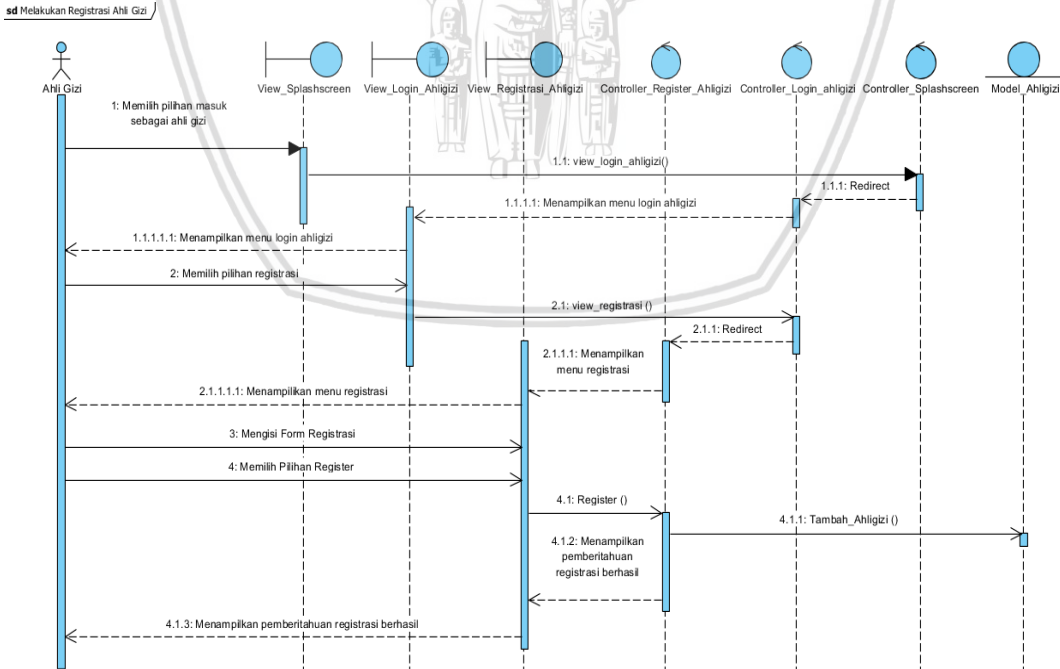
Gambar 4.13 dibawah ini merupakan deskripsi *sequence diagram* dari *use case* Melakukan Logout yang dilakukan oleh pengguna.



Gambar 4.13 Sequence Diagram Melakukan Logout

4.3.12 Sequence Diagram Melakukan Registrasi Ahli Gizi

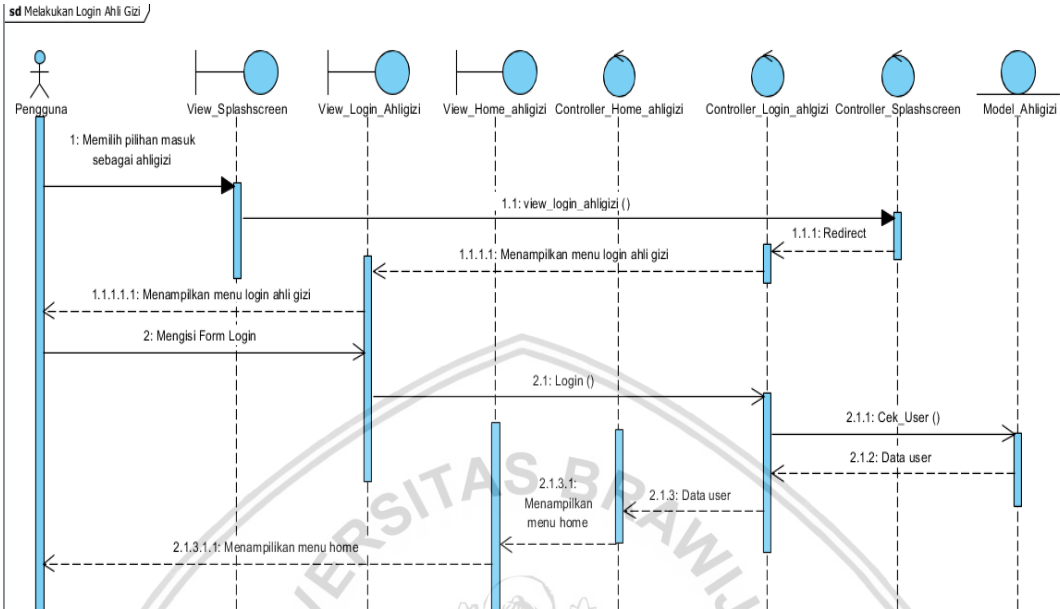
Gambar 4.14 dibawah ini merupakan deskripsi *sequence diagram* dari *use case* Melakukan Registrasi yang dilakukan oleh ahli gizi.



Gambar 4.14 Sequence Diagram Melakukan Login Ahli Gizi

4.3.13 Sequence Diagram Melakukan Login Ahli Gizi

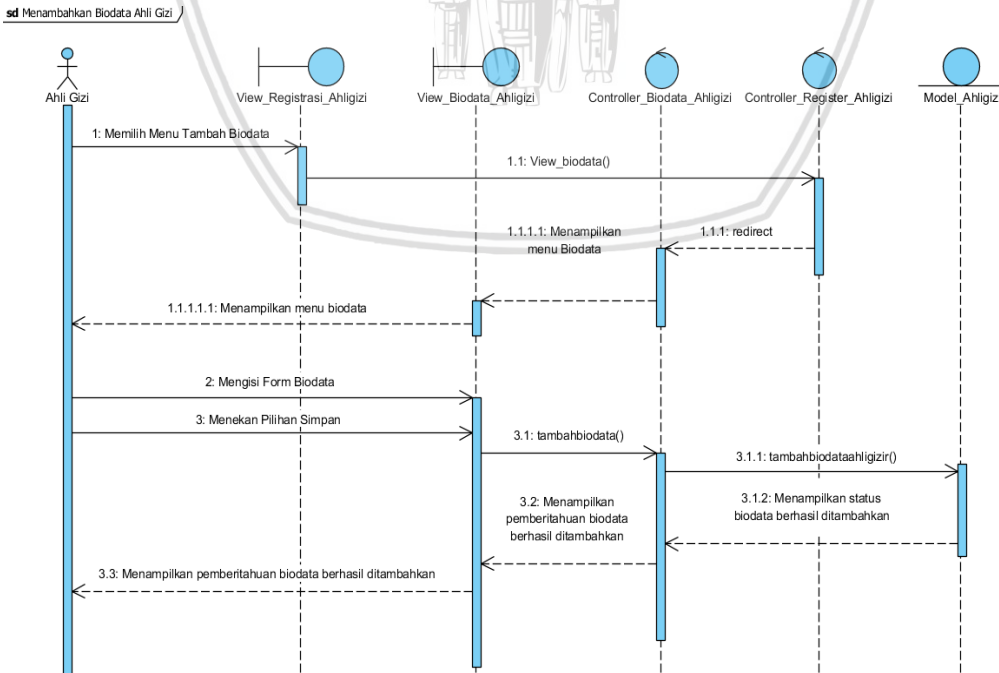
Gambar 4.15 dibawah ini merupakan deskripsi *sequence diagram* dari *use case* Melakukan Login yang dilakukan oleh ahli gizi.



Gambar 4.15 Sequence Diagram Melakukan Registrasi Ahli Gizi

4.3.14 Sequence Diagram Menambahkan Biodata Ahli Gizi

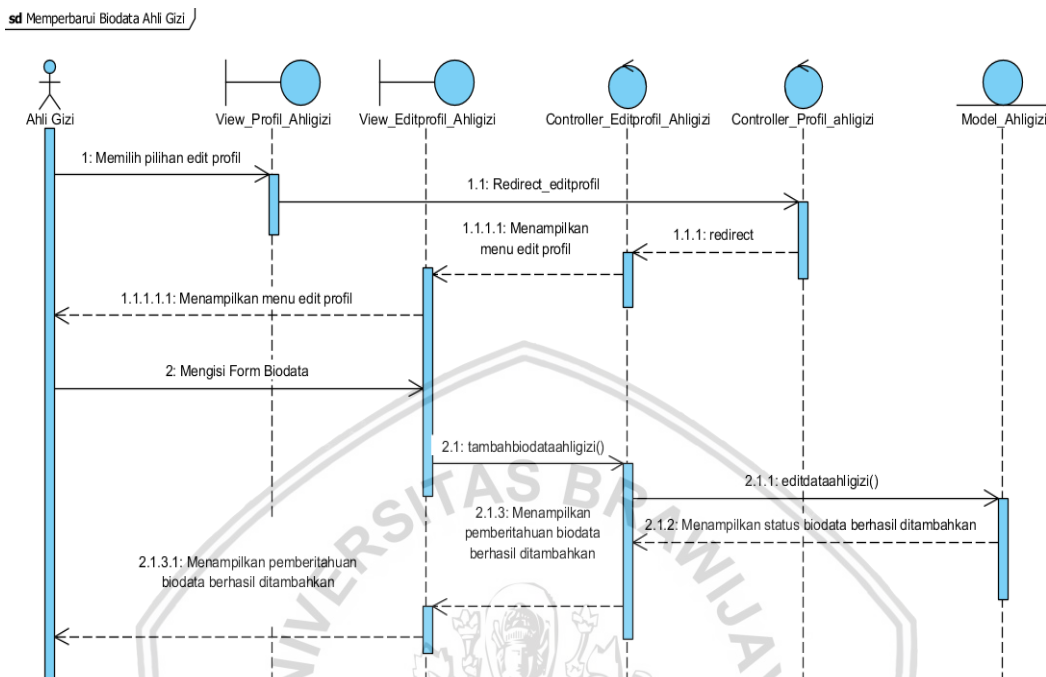
Gambar 4.16 dibawah ini merupakan deskripsi *sequence diagram* dari *use case* Menambahkan Biodata yang dilakukan oleh ahli gizi.



Gambar 4.16 Sequence Diagram Menambahkan Biodata Ahli Gizi

4.3.15 Sequence Diagram Memperbarui Biodata Ahli Gizi

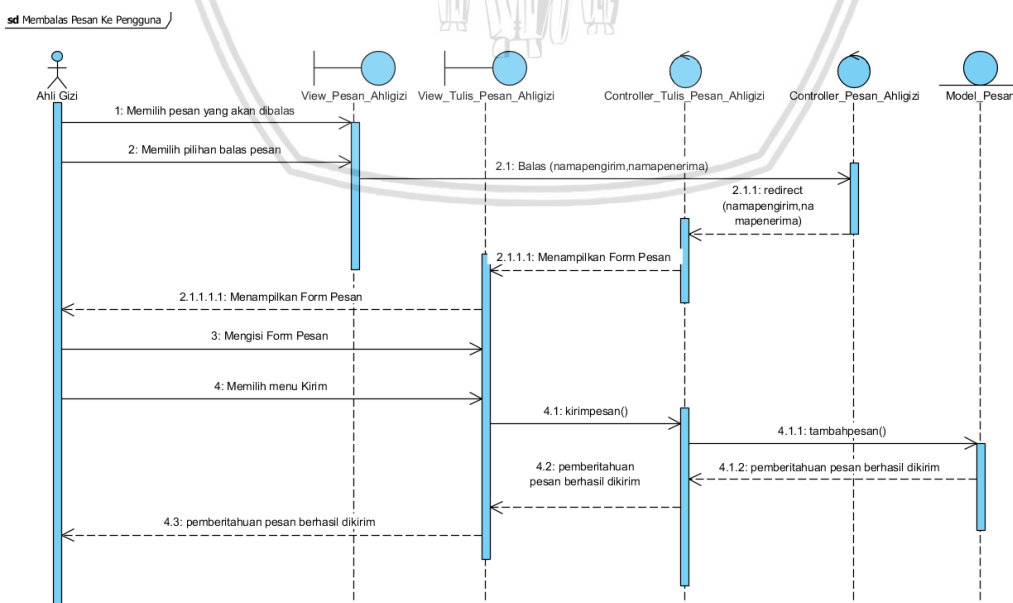
Gambar 4.17 dibawah ini merupakan deskripsi *sequence diagram* dari *use case* Memperbarui Biodata yang dilakukan oleh ahli gizi.



Gambar 4.17 Sequence Diagram Memperbarui Biodata Ahli Gizi

4.3.16 Sequence Diagram Membalas Pesan Ke Pengguna

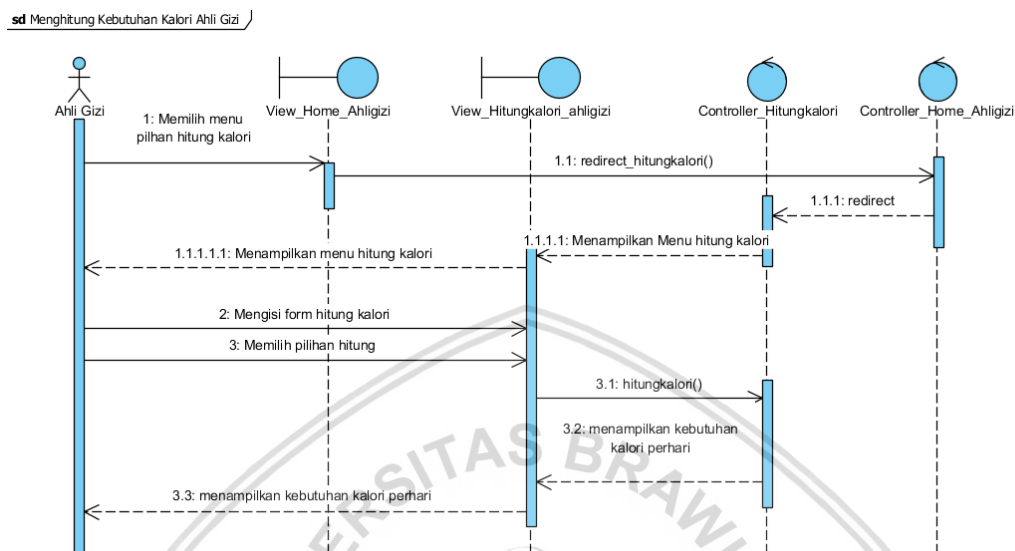
Gambar 4.18 dibawah ini merupakan deskripsi *sequence diagram* dari *use case* Membalas Pesan ke Pengguna yang dilakukan oleh ahli gizi.



Gambar 4.18 Sequence Diagram Membalas Pesan Ke Pengguna

4.3.17 Sequence Diagram Menghitung Kebutuhan kalori Perhari

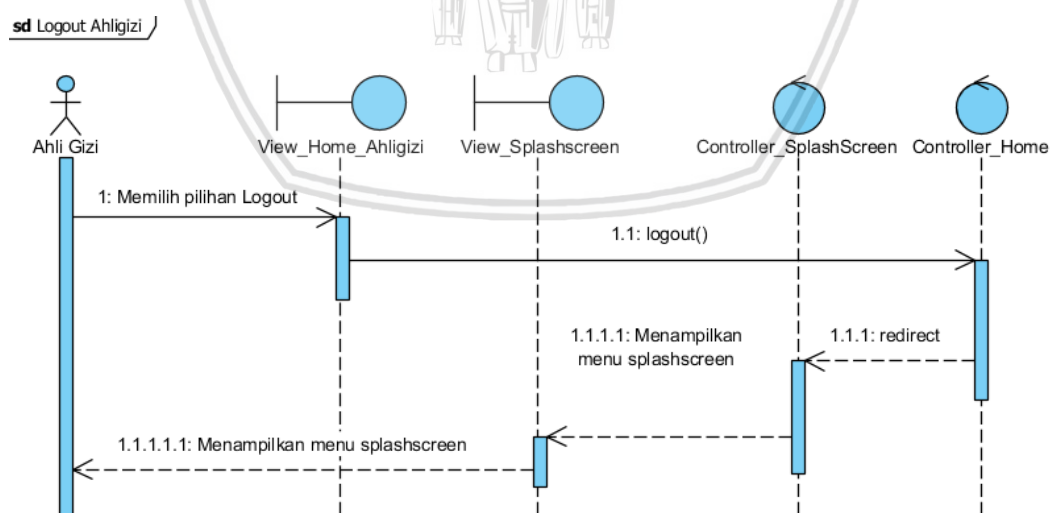
Gambar 4.19 dibawah ini merupakan deskripsi *sequence diagram* dari use case Membalas Pesan ke Pengguna yang dilakukan oleh ahli gizi.



Gambar 4.19 Sequence Diagram Menghitung Kebutuhan Kalori Perhari

4.3.18 Sequence Diagram Melakukan Logout

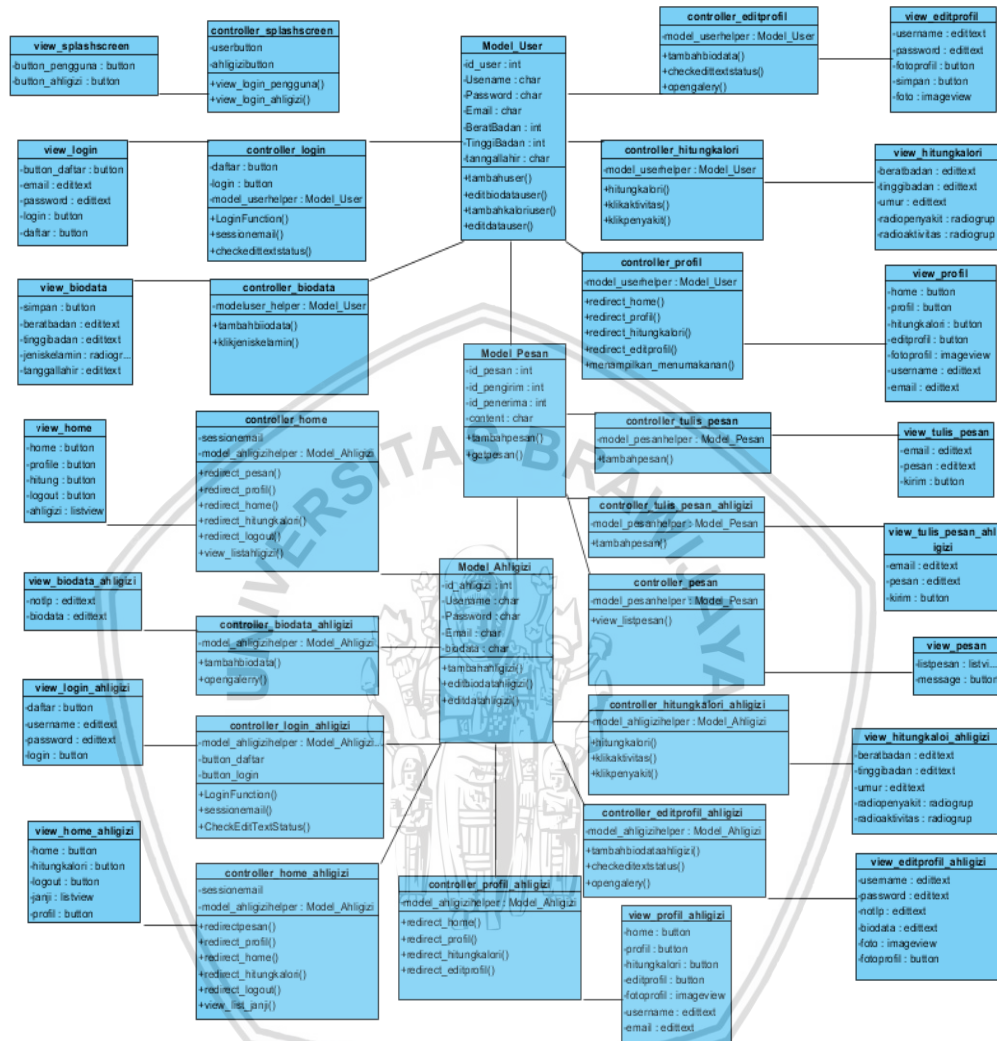
Gambar 4.20 dibawah ini merupakan deskripsi *sequence diagram* dari use case Melakukan Logout yang dilakukan oleh ahli gizi.



Gambar 4.20 Sequence Diagram Melakukan Logout

4.4 Class Diagram

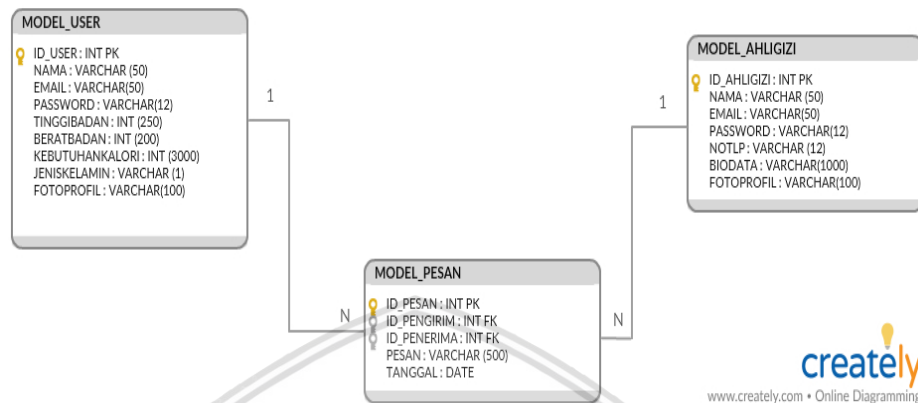
Class diagram dibuat berdasarkan perancangan yang telah dibuat sebelumnya. Pada gambar 4.21 dibawah ini merupakan gambaran dari class diagram pada aplikasi yang akan dibangun.



Gambar 4.21 Class Diagram Aplikasi.

4.5 Perancangan *Physical Data Model*

Pada subbab ini akan digambarkan secara umum bagaimana perancangan data yang akan diimplementasikan pada aplikasi nantinya. Gambar 4.22 merupakan deskripsi dari *physical data model* dari aplikasi yang akan dibuat



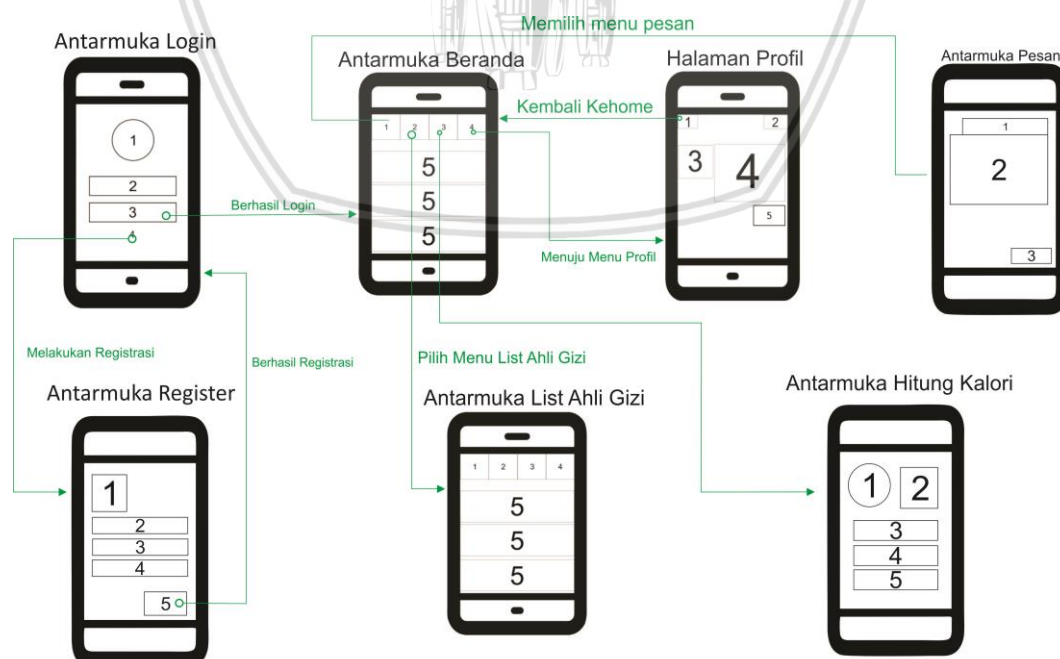
Gambar 4.22 *Physical Data Model*

4.6 Perancangan Antar Muka Aplikasi

Pada subbab ini akan digambarkan secara umum bagaimana tampilan aplikasi nantinya dibuat, serta bayangan *user interface* dan tombol" yang ada di aplikasi.

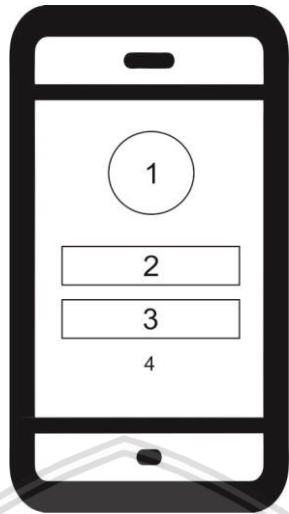
4.6.1 Screen Flow

Pada gambar 4.23 dideskripsikan bagaimana perancangan *screen flow* atau alur tampilan pada sistem nantinya



Gambar 4.23 *Screen Flow* Aplikasi

4.6.2 Antar Muka *Login*



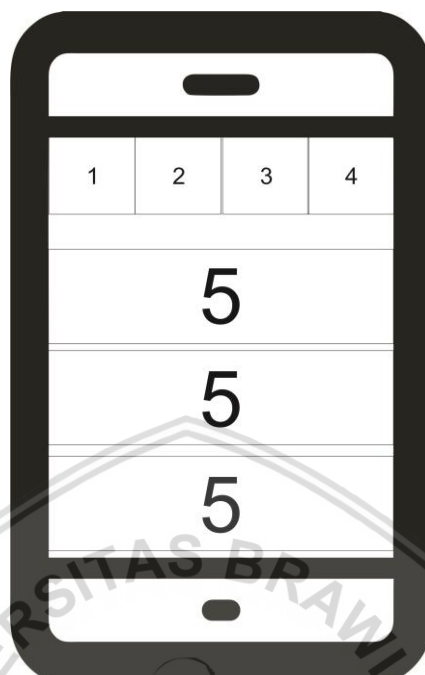
Gambar 4.24
Tampilan *Login*

Penjelasan : tempat melakukan proses *Login* dan register.

Tabel 4.68 Tabel Keterangan Antar Muka *Login*

| No | Nama objek | Tipe | Keterangan |
|----|-----------------------|-----------|--|
| 1 | Gambar Logo | Image | Logo dari Aplikasi |
| 2 | <i>Username</i> Field | TextField | Tempat memasukan nama <i>username</i> untuk <i>login</i> |
| 3 | Password Field | TextField | Tempat memasukan password untuk <i>login</i> |
| 4 | RegisterLink | Link | Link untuk melakukan registrasi |

4.6.3 Antar Muka Beranda



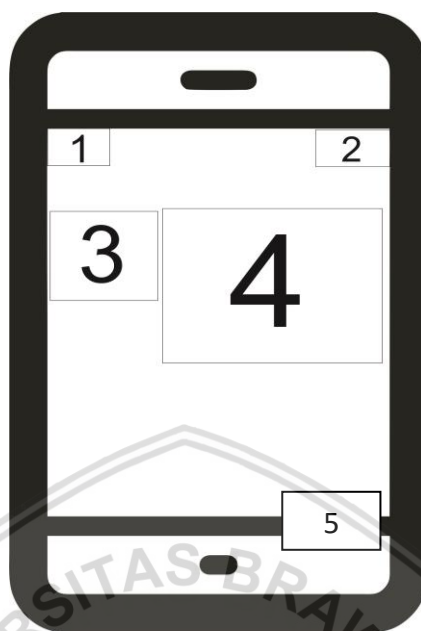
Gambar 4.25 Antar Muka Beranda

Penjelasan: halaman utama aplikasi yang berisi pilihan - pilihan fitur yang dapat dilakukan oleh *user*.

Tabel 4.69 Tabel Antar Muka Beranda

| No | Nama objek | Tipe | Keterangan |
|----|------------------------------|--------|---|
| 1 | Tombol Profile | Tombol | Tombol untuk melihat Profil |
| 2 | Tombol <i>List</i> Ahli Gizi | Tombol | Tombol untuk melihat <i>List</i> ahli gizi |
| 3 | Tombol Hitung Kalori | Tombol | Tombol untuk melakukan perhitungan kalori yang dibutuhkan tubuh |
| 4 | Tombol Pesan | Tombol | Tombol untuk melihat atau menulis pesan |

4.6.4 Antar Muka Profil



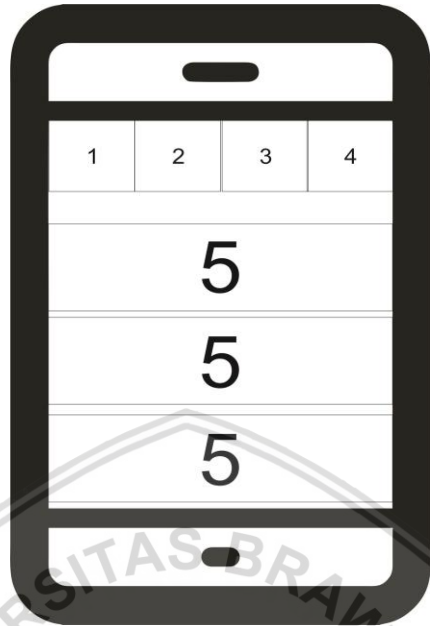
Gambar 4.26 Antar Muka Profil

Penjelasan: halaman utama aplikasi yang berisi pilihan - pilihan fitur yang dapat dilakukan oleh *user*.

Tabel 4.70 Tabel Antar Muka Profil

| No | Nama objek | Tipe | Keterangan |
|----|---------------------|------------|--|
| 1 | Tombol Home | Tombol | Tombol untuk Kembali Ke Profil |
| 2 | Tombol Sign out | Tombol | Tombol untuk Sign Out dari aplikasi |
| 3 | Profil Picture | Image | Foto Profil dari <i>user</i> |
| 4 | Biodata <i>User</i> | Text Field | Biodata dari <i>user</i> yang berisi berat badan ,tinggi , nama serta kalori yang ia butuhkan perharinya |
| 5 | Tombo Edit Profile | Tombol | Tombol Untuk Edit Profile |

4.6.5 Antar Muka Lihat Ahli Gizi



Gambar 4.27 Antar Muka Lihat List ahli gizi

Penjelasan: antar muka dari fitur melihat list ahli gizi

Tabel 4.71 Tabel Antar Lihat Pesan

| No | Nama objek | Tipe | Keterangan |
|----|------------------------------|--------|---|
| 1 | Tombol Profile | Tombol | Tombol untuk melihat Profil |
| 2 | Tombol <i>List</i> Ahli Gizi | Tombol | Tombol untuk melihat <i>List</i> ahli gizi |
| 3 | Tombol Hitung Kalori | Tombol | Tombol untuk melakukan perhitungan kalori yang dibutuhkan tubuh |
| 4 | Tombol Pesan | Tombol | Tombol untuk melihat atau menulis pesan |

BAB 5 IMPLEMENTASI DAN PEMBAHASAN

Implementasi aplikasi konsultasi menu makanan bagi penyandang diabetes in memanfaatkan metode *harris benedict* sebagai rumus dasar untuk menentukan kalori yang dibutuhkan penderita perharinya. Dimana metode *harris benedict* dituangkan didalam kode program sehingga aplikasi dapat menghitung kalori *user* secara tepat. Implementasi aplikasi perangkat bergerak ini menggunakan bahasa java, database SQLite dan tools pendukung lainnya, Implementasi meliputi:

- Pembuatan *user interface*
- Menerapkan metode *harris benedict* dalam aplikasi yang telah dibuat menggunakan bahasa java

Output berupa laporan kalori yang dibutuhkan *user* serta saran menu makanan yang cocok dengan kalori *user* tersebut,

5.1 Implementasi Fitur Melakukan Registrasi

Penerapan fitur registrasi bermanfaat untuk menangkap biodata pengguna yang sebagai acuan penghitungan kalori menggunakan metode *haris bendict*. Pada aplikasi perangkat bergerak ini penerapan fitur registrasi terletak pada *class* controller_registrasi. Tabel 5.1 berikut merupakan implementasi dari fitur *register* yang terdapat pada *class* controller_register.

Tabel 5.1 *Source Code* Implementasi Fitur Register

| No | Source Code |
|----|---|
| 1 | <code>public class controller_register extends AppCompatActivity {</code> |
| 2 | <code>private SQLiteDatabase sqLiteDatabaseObj;</code> |
| 3 | <code>String sqLiteDataBaseQueryHolder ;</code> |
| 4 | <code>private model_user model_userhelper;</code> |
| 5 | <code>Cursor cursor;</code> |
| 6 | <code>String F_Result = "Not_Found";</code> |
| 7 | <code>String inputnama,inputemail,inputpassword;</code> |
| 8 | <code>EditText holdernama,holderemail,holderpassword;</code> |
| 9 | <code>Button register;</code> |
| 10 | <code>Boolean EditTextEmptyHolder;</code> |
| 11 | |
| 12 | <code>@Override</code> |
| 13 | <code>protected void onCreate(Bundle savedInstanceState) {</code> |
| 14 | <code>super.onCreate(savedInstanceState);</code> |
| 15 | <code>setContentView(R.layout.view_register);</code> |
| 16 | |
| 17 | <code>//cari inputan register</code> |
| 18 | <code>holdernama =(EditText) findViewById(R.id.nama);</code> |
| 19 | <code>holderemail=(EditText) findViewById(R.id.email);</code> |
| 20 | <code>holderpassword=(EditText) findViewById(R.id.password);</code> |
| 21 | <code>register = (Button) findViewById(R.id.register);</code> |
| 22 | <code>model_userhelper = new model_user(this);</code> |
| 23 | |
| 24 | |
| 25 | <code>register.setOnClickListener(new View.OnClickListener() {</code> |
| 26 | |
| 27 | <code>public void onClick(View view) {</code> |
| 28 | |
| 29 | |
| 30 | |

```

31      // Creating SQLite database if dose n't exists
32      SQLiteDataBaseBuild();
33
34      // Creating SQLite table if dose n't exists.
35      SQLiteTableBuild();
36
37      // Checking EditText is empty or Not.
38      CheckEditTextStatus();
39
40      // Method to check Email is already exists or not.
41      CheckingEmailAlreadyExistsOrNot();
42
43      // Empty EditText After done inserting process.
44      EmptyEditTextAfterDataInsert();
45
46
47      }
48      });
49  }
50
51  public void SQLiteDataBaseBuild() {
52
53      SQLiteDatabaseObj =
54      openOrCreateDatabase(model_userhelper.DATABASE_NAME,
55      Context.MODE_PRIVATE, null);
56      }
57
58
59  public void SQLiteTableBuild() {
60
61      SQLiteDatabaseObj.execSQL("CREATE TABLE IF NOT EXISTS " +
62      model_userhelper.TABLE_NAME + "("
63      + model_userhelper.Table_Column_ID + " INTEGER
64      PRIMARY KEY AUTOINCREMENT NOT NULL, "
65      + model_userhelper.Table_Column_1_nama + " VARCHAR,
66      "
67      + model_userhelper.Table_Column_2_email + "
68      VARCHAR, "
69      + model_userhelper.Table_Column_3_password+"
70      VARCHAR, "
71      + model_userhelper.Table_Column_4_tinggibadan+"
72      INTEGER, "
73      + model_userhelper.Table_Column_5_beratbadan+"
74      INTEGER, "
75      + model_userhelper.Table_Column_6_tanggallahir+"
76      DATE, "
77      + model_userhelper.Table_Column_7_kebutuhankalori+"
78      INTEGER );");
79
80      }
81
82
83  public void CheckEditTextStatus() {
84
85      // Getting value from All EditText and storing into String
86      Variables.
87      inputnama = holdernama.getText().toString() ;
88      inputemail = holderemail.getText().toString();
89      inputpassword = holderpassword.getText().toString();
90
91      if(TextUtils.isEmpty(inputnama) ||
92      TextUtils.isEmpty(inputemail) || TextUtils.isEmpty(inputpassword)){
93          EditTextEmptyHolder = false ;
94      }
95      else {

```

```

96         EditTextEmptyHolder = true ;
97     }
98 }
99
100 public void CheckingEmailAlreadyExistsOrNot() {
101
102     // Opening SQLite database write permission.
103     sqLiteDatabaseObj = model_userhelper.getWritableDatabase();
104
105     // Adding search email query to cursor.
106     cursor =
107     sqLiteDatabaseObj.query(model_userhelper.TABLE_NAME, null, " " +
108     model_userhelper.Table_Column_2_email + "=?", new
109     String[]{inputemail}, null, null, null, null);
110
111     while (cursor.moveToNext()) {
112         if (cursor.isFirst()) {
113             cursor.moveToFirst();
114             // If Email is already exists then Result variable
115             value set as Email Found.
116             F_Result = "Email Found";
117
118             // Closing cursor.
119             cursor.close();
120         }
121     }
122
123     // Calling method to check final result and insert data
124     into SQLite database.
125     CheckFinalResult();
126
127 }
128
129 public void CheckFinalResult() {
130
131     // Checking whether email is already exists or not.
132     if(F_Result.equalsIgnoreCase("Email Found"))
133     {
134         // If email is exists then toast msg will display.
135         //
136         model_userhelper.deleteuser(sqLiteDatabaseObj,inputnama);
137         Toast.makeText(controller_register.this,"Email Already
138         Exists",Toast.LENGTH_LONG).show();
139     }
140
141     else{
142         // If email already dose n't exists then user
143         registration details will entered to SQLite database.
144
145         if(EditTextEmptyHolder == true)
146         {
147             // SQLite query to insert data into table.
148
149             model_userhelper.tambahuser(sqLiteDatabaseObj,inputnama,inputemail,
150             inputpassword);
151
152
153             //pemberitahuan register berhasil
154             Toast.makeText(controller_register.this,"Registrasi
155             berhasil.",Toast.LENGTH_LONG).show();
156             Intent intent = new
157             Intent(controller_register.this,controller biodata.class);
158             intent.putExtra("UserEmail", inputemail);
159             startActivity(intent);
160

```

```

161
162         }
163         // This block will execute if any of the registration
164         EditText is empty.
165         else {
166             // pemberitahuan inputan kosong
167             Toast.makeText(controller_register.this, "Data anda
168             masih kosong.", Toast.LENGTH_LONG).show();
169         }
170
171     }
172     F_Result = "Not_Found" ;
173 }
174 }
175 }
176

```

Pada baris 25 adalah fungsi yang berjalan ketika tombol register ditekan oleh pengguna. Didalam fungsi tersebut terdapat beberapa fungsi yang akan berjalan yaitu fungsi *SQLiteDataBaseBuild* yang dipanggil pada baris ke 32 fungsi *SQLiteTableBuild* yang dipanggil pada baris ke 35, fungsi *CheckEditTextStatus* yang dipanggil pada baris ke 38 dan fungsi *CheckingEmailAlreadyExistOrNot* yang dipanggil pada baris ke 41.

kemudian fungsi *DatDataBaseBuild* terletak pada baris 50 diatas, dimana fungsi tersebut berguna untuk membuat databas objek dari model *_user*. Lalu fungsi *SQLiteTableBuild* terletak pada baris 61 diatas, dimana fungsi tersebut berguna untuk membuat tabel *user* apabila tabel tersebut belum terdapat di aplikasi.

Fungsi *CheckEditTextStatus* yang terdapat pada baris 83 berguna untuk memeriksa apakah *user* telah mengisi *form* registrasi dengan benar. Apabila terdapat kolom yang kosong maka fungsi ini akan memberitahu pengguna tersebut bahwa ada beberapa data yang belum dilengkapi.

Dan fungsi *CheckingEmailAlreadyExistOrNot* yang terdapat pada baris 100 berguna untuk mengecek apakah email yang dimasukan pengguna didalam *form* registrasi sudah pernah dipakai *user* lain atau belum. Apabila sudah , maka *user* akan diberikan pemberitahuan bahwa email tersebut pernah dipakai.

5.2 Implementasi Fitur Melakukan *Login*

Implementasi fitur *login* pada aplikasi perangkat bergerak ini berguna untuk melakukan autentikasi pengguna sehingga data yang diberikan aplikasi ini nantinya sesuai dengan biodata *user* yang telah diberikan saat melakukan registrasi. Implementasi fitur *login* pada aplikasi ini dilakukan pada *class* *controller_login*. Tabel 5.2 dibawah ini merupakan *Source Code* dari implementasi fitur *login* pada *class* *controller_login*.

Tabel 5.2 Source Code Implementasi Fitur Login

| No | Source Code |
|----|--|
| 1 | public void LoginFunction() { |
| 2 | |
| 3 | if (EditTextEmptyHolder) { |
| 4 | |
| 5 | <i>// Opening SQLite database write permission.</i> |
| 6 | sqliteDatabaseObj = model_userhelper.getWritableDatabase(); |
| 7 | |
| 8 | <i>// Adding search email query to cursor.</i> |
| 9 | cursor = |
| 10 | sqliteDatabaseObj.query(model_userhelper.TABLE_NAME, null, " " + |
| 11 | model_userhelper.Table_Column_2_email + "=?", new |
| 12 | String[]{inputanemail}, null, null, null); |
| 13 | |
| 14 | while (cursor.moveToNext()) { |
| 15 | |
| 16 | if (cursor.isFirst()) { |
| 17 | |
| 18 | cursor.moveToFirst(); |
| 19 | |
| 20 | <i>// Storing Password associated with entered email.</i> |
| 21 | TempPassword = |
| 22 | cursor.getString(cursor.getColumnIndex(model_userhelper.Table_Column |
| 23 | _3_password)); |
| 24 | |
| 25 | <i>// Closing cursor.</i> |
| 26 | cursor.close(); |
| 27 | } |
| 28 | } |
| 29 | |
| 30 | <i>// Calling method to check final result ..</i> |
| 31 | CheckFinalResult(); |
| 32 | |
| 33 | } else { |
| 34 | |
| 35 | <i>//If any of login EditText empty then this block will be</i> |
| 36 | <i>executed.</i> |
| 37 | Toast.makeText(controller_login.this, "Tolong masukan email |
| 38 | dan password yang tepat.", Toast.LENGTH_LONG).show(); |
| 39 | |
| 40 | } |
| 41 | |
| 42 | } |
| 43 | |
| 44 | public void CheckFinalResult() { |
| 45 | |
| 46 | if (TempPassword.equalsIgnoreCase(inputanpassword)) { |
| 47 | |
| 48 | Toast.makeText(controller_login.this, "Login Successfully", |
| 49 | Toast.LENGTH_LONG).show(); |
| 50 | session = getSharedPreferences(MyPREFERENCES, |
| 51 | Context.MODE_PRIVATE); |
| 52 | SharedPreferences.Editor editor = session.edit(); |
| 53 | editor.putString("EmailSession", inputanemail); |
| 54 | editor.commit(); |
| 55 | |
| 56 | sessionemail(); |
| 57 | <i>// Going to Dashboard activity after login success message.</i> |
| 58 | Intent intent = new Intent(controller_login.this, |
| 59 | controller_home.class); |
| 60 | startActivity(intent); |
| 61 | |
| 62 | } else { |

| | |
|----|--|
| 63 | <code>cobal.setText(TempPassword);</code> |
| 64 | <code>Toast.makeText(controller_login.this, "UserName or Password</code> |
| 65 | <code>is Wrong, Please Try Again.", Toast.LENGTH_LONG).show();</code> |
| 66 | |
| 67 | <code>}</code> |
| 68 | <code>}</code> |

Baris ke 9 sampai dengan 12 merupakan *query* untuk melakukan pengecekan apakah email yang dimasukan *user* terdaftar pada database atau tidak. Lalu dari pengecekan email tersebut dilakukan pengecekan password pada fungsi *CheckFinalResult* yang dipanggil pada baris 31.

Pada baris ke 44 dilakukan pengecekan password dari *user* dengan password yang terdaftar pada database. Apabila password tersebut benar maka *user* akan langsung diarahkan ke halaman home.

5.3 Implementasi Fitur Menambahkan Biodata

Implementasi fitur masukan biodata pada aplikasi perangkat bergerak ini berguna untuk menerima biodata *user* seperti tinggi badan, berat badan dan tanggal lahir. Dimana biodata tersebut bermanfaat sebagai parameter untuk melakukan perhitungan kalori menggunakan metode *harrise bendict* nantinya. Implementasi fitur masukan biodata pada aplikasi perangkat bergerak ini dilakukan pada *class* *controller_biodata*. Tabel 5.3 dibawah ini merupakan *source code* dari implementasi fitur masukan biodata.

Tabel 5.3 Source Code Implementasi Fitur Tambah Biodata

| No | Source Code |
|----|--|
| 1 | <code>public void tambahbiodata(){</code> |
| 2 | <code> SQLiteDataBaseBuild();</code> |
| 3 | <code> int tinggibadan =</code> |
| 4 | <code>Integer.parseInt(holdertinggibadan.getText().toString());</code> |
| 5 | <code> int beratbadan =</code> |
| 6 | <code>Integer.parseInt(holderberatbadan.getText().toString());</code> |
| 7 | |
| 8 | <code> sqliteDatabaseObj = model_userhelper.getWritableDatabase();</code> |
| 9 | <code>model_userhelper.tambahbiodatauser(sqliteDatabaseObj,email,</code> |
| 10 | <code>beratbadan,tinggibadan,date);</code> |
| 11 | <code> Intent intent = new Intent(controller_biodata.this,</code> |
| 12 | <code>controller_login.class);</code> |
| 13 | <code> startActivity(intent);</code> |
| 14 | |
| 15 | <code>}</code> |

Pada baris 4 sampai 6 dilakukan pengambilan inputan tinggi badan dan berat badan dari *user*. Kemudian dilakukan pemanggilan fungsi *tambahbiodatauser* yang terdapat pada *class* *model_user* dengan parameter email, berat badan, tinggi badan dan tanggal lahir *user*.

5.4 Implementasi Fitur Memperbarui Biodata

Implementasi fitur memperbarui biodata pada aplikasi perangkat bergerak ini berguna untuk melakukan pemberauman biodata pada aplikasi, apabila *user* ingin melakukan pemberauman biodata pada aplikasi. Pada tabel 5.4 merupakan implementasi *source code* dari fitur memperbarui biodata.

Tabel 5.4 Source Code Implementasi Fitur Memperbarui Biodata

| No | Source Code |
|----|--|
| 1 | <code>public void tambahbiodata() {</code> |
| 2 | <code> SQLiteDataBaseBuild();</code> |
| 3 | <code> int tinggibadan =</code> |
| 4 | <code>Integer.parseInt(holdertinggibadan.getText().toString());</code> |
| 5 | <code> int beratbadan =</code> |
| 6 | <code>Integer.parseInt(holderberatbadan.getText().toString());</code> |
| 7 | |
| 8 | <code> sqliteDatabaseObj = model_userhelper.getWritableDatabase();</code> |
| 9 | <code>model_userhelper.tambahbiodatauser(sqliteDatabaseObj, email,</code> |
| 10 | <code>beratbadan, tinggibadan, date);</code> |
| 11 | <code> Intent intent = new Intent(controller_biodata.this,</code> |
| 12 | <code>controller_login.class);</code> |
| 13 | <code> startActivity(intent);</code> |
| 14 | |
| 15 | <code>}</code> |

Pada baris 4 sampai 6 dilakukan pengambilan inputan tinggi badan dan berat badan dari *user*. Kemudian dilakukan pemanggilan fungsi *tambahbiodatauser* yang terdapat pada *class* *model_user* dengan parameter email, berat badan, tinggi badan dan tanggal lahir *user*.

5.5 Implementasi Fitur Memperbarui Profil

Implementasi fitur memperbarui biodata pada aplikasi perangkat bergerak ini berguna untuk melakukan pemberauman profil pada aplikasi, apabila *user* ingin memperbarui profil pada aplikasi. Pada tabel 5.5 merupakan implementasi *source code* dari fitur memperbarui biodata.

Tabel 5.5 Source Code Implementasi Fitur Memperbarui Profil

| No | Source Code |
|----|--|
| 1 | <code>SQLiteDataBaseBuild();</code> |
| 2 | <code>int tinggibadan =</code> |
| 3 | <code>Integer.parseInt(holdertinggibadan.getText().toString());</code> |
| 4 | <code>int beratbadan =</code> |
| 5 | <code>Integer.parseInt(holderberatbadan.getText().toString());</code> |
| 6 | <code>passwordbaru = holderpassword.getText().toString();</code> |
| 7 | <code>namabaru = holderusername.getText().toString();</code> |
| 8 | <code>sqliteDatabaseObj = model_userhelper.getWritableDatabase();</code> |
| 9 | |
| 10 | <code>model_userhelper.editdatauser(sqliteDatabaseObj, email, beratbadan, t</code> |
| 11 | <code>inggibadan, date, jeniskelamin, namabaru, passwordbaru, linkfoto);</code> |
| 12 | |
| 13 | <code>Intent intent = new Intent(controller_editprofil.this,</code> |
| 14 | <code>controller_profil.class);</code> |
| 15 | <code>SharedPreferences aba =</code> |
| 16 | <code>getSharedPreferences(controller_login.MyPREFERENCES,</code> |
| 17 | <code>Context.MODE_PRIVATE);</code> |
| 18 | <code>startActivity(intent);</code> |
| 19 | |

Pada baris 2 sampai 7 dilakukan pengambilan inputan *username*, *password*, *tinggi badan* dan *berat badan* dari *user*. Kemudian dilakukan pemanggilan fungsi *editdatauser* yang terdapat pada *class model_user* dengan parameter *email*, *berat badan*, *tinggi badan* dan *tanggal lahir user*.

5.6 Implementasi Fitur Menghitung Kebutuhan Kalori perhari

Implementasi fitur hitung kalori pada aplikasi perangkat bergerak ini berguna untuk menentukan jumlah kalori yang dibutuhkan *user* menggunakan metode *harris benedict*. Pada tabel 5.6 dibawa ini merupakan *Source Code* implementasi fitur hitung kalori yang terdapat pada *class controller_hitungkalori*.

Tabel 5.6 Source Code Implementasi Fitur Hitung Kalori

| No | Source Code |
|----|---|
| 1 | <code>public void hitungkalori (View view){</code> |
| 2 | <code> nilaiberatbadan =</code> |
| 3 | <code>Integer.parseInt(beratbadan.getText().toString());</code> |
| 4 | <code> nilaitinggibadan =</code> |
| 5 | <code>Integer.parseInt(tinggibadan.getText().toString());</code> |
| 6 | <code> nilaiumur = Integer.parseInt(umur.getText().toString());</code> |
| 7 | <code> //metodeharrisbenedict</code> |
| 8 | <code> double AMB = 655+ (9.6 * nilaiberatbadan) + (1.8 *</code> |
| 9 | <code>nilaitinggibadan) - (4.7 * nilaiumur);</code> |
| 10 | <code> double energi = AMB * nilaistress * nilaiaktivitas;</code> |
| 11 | <code> hasil = String.valueOf(energi);</code> |
| 12 | <code> hitung.setText(hasil);</code> |
| 13 | <code> int i= (int) Math.round(energi);</code> |
| 14 | <code> model_userhelper.tambahkaloriuser(sqliteDatabaseObj2</code> |
| 15 | <code>,i,sessionemail);</code> |
| 16 | <code> Intent intent = new Intent(controller_hitungkalori.this,</code> |
| 17 | <code>controller_profil.class);</code> |
| 18 | <code> startActivity(intent);</code> |
| 19 | <code>}</code> |

Pada baris ke 3 sampai ke 6 merupakan pengambilan data berat badan, tinggi badan dan umur dari inputan *user*. Lalu pada baris 8 dimulailah penghitungan kalori menggunakan

5.7 Implementasi Fitur Melihat Saran Menu Makanan

Implementasi fitur menampilkan saran menu makanan pada aplikasi perangkat bergerak ini berguna untuk memberikan saran menu makanan bagi *user* sesuai dengan kalori yang telah dihitung sebelumnya. Pada tabel 5.7 dibawa ini merupakan *Source Code* implementasi fitur menampilkan saran menu makanan yang terdapat pada *class controller_profil*.

Tabel 5.7 Source Code Implementasi Ftur Menampilkan Saran Menu Makanan

| No | Source Code |
|----|---|
| 1 | <code>public void menampilkan_menumakanan(int kalori, String jk) {</code> |
| 2 | |
| 3 | <code> //jika jenis kelamin laki laki</code> |
| 4 | |
| 5 | <code> if ((kalori >= 2725) && (jk.equals("L"))){</code> |
| 6 | <code> makanpagi1.setText("Bubur Tawar 200gr");</code> |
| 7 | <code> makanpagi2.setText("Sop Bayam 150gr");</code> |
| 8 | <code> makanpagi3.setText("Telur Dadar 40gr");</code> |

```

9      makanpagi4.setText("Apel Merah 140gr");
10     makanpagi5.setText("Teh 140 ml");
11
12     selinganpagi1.setText("Pisang 125gr");
13
14     makansiang1.setText("Nasi Putih 100gr");
15     makansiang2.setText("Telur Ayam Rebus 60gr");
16     makansiang3.setText("Sop Mutiara Jagung 200gr");
17     makansiang4.setText("Apel 320 gr");
18     makansiang5.setText("Pir 440 gr");
19
20     selingansore1.setText("Kue Ape 210gr");
21
22     makanmalam1.setText("Nasi Putih 100gr");
23     makanmalam2.setText("Ayam Rebus 100gr");
24     makanmalam3.setText("Melon 200 gr");
25     makanmalam4.setText("Teh Cangkir 140 ml");
26
27     selinganmalam1.setText("Susu 100gr");
28     selinganmalam2.setText("Roti Tawar 100gr");
29
30     if (kalori > 2725) {
31         saran.setText("Anda harus melakukan konsultasi
32 dengan ahli gizi untuk menu makanan yang lebih cocok");
33     }
34 }
35 //menumakanan kalori kurangdari 2725 lebih dari 2625
36 if ((kalori > 2625) && (kalori < 2725) && (jk.equals("L")))
37 {
38     makanpagi1.setText("jagung Rebus 312 gr");
39     makanpagi2.setText("Pisang Hijau 200 gr");
40     makanpagi3.setText("Telur Dadar 40gr");
41     makanpagi4.setText("Apel 140gr");
42     makanpagi5.setText("Teh Cangkir 150 ml");
43
44     selinganpagi1.setText("Pir 440gr");
45     selinganpagi1.setText("Teh Tawar 100 ml");
46
47     makansiang1.setText("Nasi Putih 100gr");
48     makansiang2.setText("Telur Ayam Rebus 60gr");
49     makansiang3.setText("Ikan Teri 50 gr");
50     makansiang4.setText("Tumis Buncis 100 gr");
51     makansiang5.setText("Pir 440 gr");
52     makansiang6.setText("Jambu Air 60 gr");
53     makansiang7.setText("Jus Melon 150 ml");
54
55     selingansore1.setText("Apel Merah 140gr");
56     selingansore2.setText("Pepaya 100gr");
57     selingansore3.setText("Jambu Biji 320gr");
58
59     makanmalam1.setText("Nasi Putih 100gr");
60     makanmalam2.setText("Ayam Rebus 100gr");
61     makanmalam3.setText("tumis Buncis 200 gr");
62     makanmalam4.setText("Teh Tawar 100 ml");
63
64     selinganmalam1.setText("Susu 100gr");
65     selinganmalam2.setText("Roti Tawar 50gr");
66     selinganmalam3.setText("Pepaya 200 gr");
67 }
68
69
70
71 if ((kalori <= 2625) && (jk.equals("L"))) {
72     makanpagi1.setText("Bubur 300 gr");
73     makanpagi2.setText("Telur Dadar 80gr");

```



```

74     makanpagi3.setText("Apel 160gr");
75     makanpagi4.setText("Teh Tawar 100 ml");
76
77     selinganpagi1.setText("Cracker 50 gr");
78     selinganpagi2.setText("Teh Tawar 100 ml");
79
80     makansiang1.setText("Nasi Putih 100gr");
81     makansiang2.setText("Telur Ayam Rebus 60gr");
82     makansiang3.setText("Ikan Teri 50 gr");
83     makansiang4.setText("Sayur Lodeh 100 gr");
84     makansiang5.setText("Pir 440 gr");
85     makansiang6.setText("Jambu Air 60 gr");
86     makansiang7.setText("Jus Melon 150 ml");
87
88     selingansore1.setText("Apel Merah 70gr");
89     selingansore2.setText("Jambu Biji 320gr");
90
91     makanmalam1.setText("Nasi Putih 100gr");
92     makanmalam2.setText("Ayam Rebus 100gr");
93     makanmalam3.setText("tumis Buncis 100 gr");
94     makanmalam4.setText("Teh Tawar 100 ml");
95
96     selinganmalam1.setText("Susu 100gr");
97     selinganmalam2.setText("Roti Tawar 50gr");
98     selinganmalam3.setText("Pepaya 200 gr");
99
100    if (kalori < 2625) {
101        saran.setText("Anda harus melakukan konsultasi
102 dengan ahli gizi untuk menu makanan yang lebih cocok");
103    }
104 }
105
106
107 if ((kalori >= 2250) && ((jk.equals("P")))) {
108     makanpagi1.setText("Bubur ayam 200gr");
109     makanpagi2.setText("Telur Dadar 80gr");
110     makanpagi3.setText("Teh Tawar 140 ml");
111
112     selinganpagi1.setText("Cracker 50 gr");
113
114     makansiang1.setText("Nasi Putih 100gr");
115     makansiang2.setText("Telur Ayam Rebus 100gr");
116     makansiang3.setText("Sayur Lodeh 100gr");
117     makansiang4.setText("Pir 440 gr");
118     makansiang5.setText("Jus Melon 150 gr");
119
120     selingansore1.setText("Cracker 50 gr");
121
122     makanmalam1.setText("Nasi Putih 100gr");
123     makanmalam2.setText("Ayam Rebus 100gr");
124     makanmalam3.setText("Tumis Buncis 50 gr");
125     makanmalam4.setText("Teh Tawar 100 ml");
126
127     selinganmalam1.setText("Roti Tawar 50gr");
128     selinganmalam2.setText("Pepaya 200gr");
129
130     if (kalori > 2250) {
131         saran.setText("Anda harus melakukan konsultasi
132 dengan ahli gizi untuk menu makanan yang lebih cocok");
133     }
134
135     }
136     //menumakanan kalori kurangdari 2725 lebih dari 2625
137     if ((kalori > 2125) && (kalori < 2250)&&
138 ((jk.equals("P")))) {

```



```

139 makanpagi1.setText("Bubur Ayam 200 gr");
140 makanpagi2.setText("Roti Tawar 50 gr");
141 makanpagi3.setText("Apel 160gr");
142 makanpagi4.setText("Jus Melon 170 ml");
143
144 selinganpagi1.setText("Pisang Rebus 125gr");
145 selinganpagi1.setText("Apel 160 ml");
146
147 makansiang1.setText("Nasi Putih 100gr");
148 makansiang2.setText("Sayur Asam 120 gr");
149 makansiang3.setText("Ikan Teri 50 gr");
150 makansiang4.setText("Perkedel Jagung 50 gr");
151 makansiang5.setText("Jus Melon 150 ml");
152
153 selingansore1.setText("Cracker 50gr");
154
155 makanmalam1.setText("Nasi Putih 70 gr");
156 makanmalam2.setText("Ayam Rebus 100gr");
157 makanmalam3.setText("Teh Tawar 100 ml");
158
159 selinganmalam1.setText("Cracker 50gr");
160 }
161
162 if (kalori <= 2125 && ((jk.equals("P")))) {
163 makanpagi1.setText("Roti Tawar 50 gr");
164 makanpagi3.setText("Telur Dadar 75 gr");
165 makanpagi4.setText("Apel 160gr");
166 makanpagi5.setText("Jus Melon 170 ml");
167
168 selinganpagi1.setText("Pisang Rebus 125 gr");
169 selinganpagi1.setText("Salak 150 ml");
170
171 makansiang1.setText("Nasi Putih 100gr");
172 makansiang2.setText("Ikan Teri 50 gr");
173 makansiang3.setText("Perkedel Jagung 50 gr ");
174 makansiang4.setText("Sop Bayam 50 gr");
175 makansiang5.setText("Jambu Air 60 gr");
176 makansiang6.setText("Jus Melon 150 ml");
177
178 selingansore1.setText("Jambu Biji 320gr");
179 selingansore2.setText("jeruk Pontianak 145");
180
181 makanmalam1.setText("Nasi Putih 100gr");
182 makanmalam2.setText("Ayam Rebus 100gr");
183 makanmalam3.setText("tumis Buncis 100 gr");
184
185 selinganmalam1.setText("Cracker 45gr");
186
187 if (kalori < 2125 && jeniskelamin == "P") {
188 saran.setText("Anda harus melakukan konsultasi
189 dengan ahli gizi untuk menu makanan yang lebih cocok");
190 }
191 }
192 }
200
201

```

Pada *source code* diatas terdapat logika yang melakukan pembagian menu makanan ketika kalori kebutuhan *user* sesuai dengan kebutuhan kalori *user* yang sudah dihitung sebelumnya.

5.8 Implementasi Fitur Melihat *List* Ahli Gizi

Implementasi fitur melihat *List* ahli gizi pada aplikasi perangkat bergerak ini berguna untuk melihat *List* ahli gizi yang sudah mendaftar pada aplikasi ini, sehingga *user* mendapatkan pilihan untuk melakukan konsultasi dengan siapa. Pada tabel 5.8 dibawa ini merupakan *Source Code* implementasi fitur *List* ahli gizi yang terdapat pada *class* controller_home.

Tabel 5.8 Source Code Melihat *List* Ahli Gizi

| No | Source Code |
|----|---|
| 1 | <code>public void view_List_ahli_gizi() {</code> |
| 2 | <code> Cursor cursor = model_ahligizihelper.getahligizi();</code> |
| 3 | |
| 4 | <code> // Allow activity to manage lifetime of the cursor.</code> |
| 5 | <code> // DEPRECATED! Runs on the UI thread, OK for small/short</code> |
| 6 | <code>queries.</code> |
| 7 | <code> startManagingCursor(cursor);</code> |
| 8 | |
| 9 | <code> // Setup mapping from cursor to view fields:</code> |
| 10 | <code> String[] fromFieldNames = new String[]</code> |
| 11 | <code> {model_ahligizihelper.Table_Column_1_nama,</code> |
| 12 | <code>model_ahligizihelper.Table_Column_2_email,</code> |
| 13 | <code>model_ahligizihelper.Table_Column_4_notlp,</code> |
| 14 | <code>model_ahligizihelper.Table_Column_5_biodata};</code> |
| 15 | <code> int[] toViewIDs = new int[]</code> |
| 16 | <code> {R.id.nama, R.id.email, R.id.notelp,</code> |
| 17 | <code>R.id.biodata};</code> |
| 18 | |
| 19 | <code> // Create adapter to map columns of the DB onto element</code> |
| 20 | <code>in the UI.</code> |
| 21 | <code> SimpleCursorAdapter myCursorAdapter =</code> |
| 22 | <code> new SimpleCursorAdapter(</code> |
| 23 | <code> this, // Context</code> |
| 24 | <code> R.layout.item_layout, // Row layout</code> |
| 25 | <code>template</code> |
| 26 | <code> cursor, // cursor (set</code> |
| 27 | <code>of DB records to map)</code> |
| 28 | <code> fromFieldNames, // DB Column</code> |
| 29 | <code>names</code> |
| 30 | <code> toViewIDs // View IDs to</code> |
| 31 | <code>put information in</code> |
| 32 | <code>);</code> |
| 33 | <code> //</code> |
| 34 | <code> // Set the adapter for the List view</code> |
| 35 | <code> ListView myList = (ListView)</code> |
| 36 | <code>findViewById(R.id.Listahligizi);</code> |
| 37 | <code> myList.setAdapter(myCursorAdapter);</code> |
| 38 | <code>}</code> |

Pada baris ke 10 dilakukan skenario pengambilan data - data ahli gizi yang terdapat di database. lalu dari data tersebut terjadi penempatan data pada masing - masing *textView* yang terdapat pada baris 16. Kemudian skenario tersebut dilakukan pada baris ke 20 dengan *item_layout* sebagai layout penampilan data - data tersebut

5.9 Implementasi Fitur Mengirim Pesan Ke Ahli Gizi

Implementasi fitur mengirim pesan ke ahli gizi pada aplikasi perangkat bergerak ini berguna bagi *user* untuk mengirim pesan ke ahli gizi, berkaitan dengan janji konsultasi yang ingin dilakukan oleh *user* nantinya. Pada tabel 5.9 dibawah ini merupakan *Source Code* implementasi mengirim pesan ke ahli gizi yang terdapat pada *class* controller_tulispesan.

Tabel 5.9 Source Code Implementasi Mengirim Pesan Ke Ahli Gizi

| No | Source Code |
|----|---|
| 1 | <code>kirim.setOnClickListener(new View.OnClickListener() {</code> |
| 2 | <code> @Override</code> |
| 3 | <code> public void onClick(View view) {</code> |
| 4 | <code> CheckingEmailAlreadyExistsOrNot();</code> |
| 5 | |
| 6 | |
| 7 | <code> };</code> |
| 8 | |
| 9 | |
| 10 | <code> }</code> |
| 11 | <code> public void kirimpesan(){</code> |
| 12 | <code> inputancontent = holdercontent.getText().toString();</code> |
| 13 | <code> // Opening new user registration activity using intent on button</code> |
| 14 | <code>click.</code> |
| 15 | <code> SQLiteDatabase sqLiteDatabaseObj =</code> |
| 16 | <code>model_janjihelper.getReadableDatabase();</code> |
| 17 | |
| 18 | <code> model_janjihelper.onCreate(sqLiteDatabaseObj);</code> |
| 19 | |
| 20 | <code>model_janjihelper.tambahjanji(sqLiteDatabaseObj, sessionemail, email, i</code> |
| 21 | <code>nputancontent);</code> |
| 22 | <code> Toast.makeText(controller_pesan.this, "Pesan Berhasil</code> |
| 23 | <code>dikirim", Toast.LENGTH_LONG).show();</code> |
| 24 | <code> Intent intent = new Intent(controller_pesan.this,</code> |
| 25 | <code>controller_home.class);</code> |
| 26 | <code> startActivity(intent);</code> |
| 27 | |
| 28 | <code> }</code> |
| 29 | |
| 30 | <code> public void CheckingEmailAlreadyExistsOrNot(){</code> |
| 31 | |
| 32 | <code> // Opening SQLite database write permission.</code> |
| 33 | <code> sqLiteDatabaseObj2 =</code> |
| 34 | <code>model_ahligizihelper.getWritableDatabase();</code> |
| 35 | |
| 36 | <code> // Adding search email query to cursor.</code> |
| 37 | <code> cursor =</code> |
| 38 | <code>sqLiteDatabaseObj2.query(model_ahligizihelper.TABLE_NAME, null, " "</code> |
| 39 | <code>+ model_ahligizihelper.Table_Column_2_email + "=?", new</code> |
| 40 | <code>String[]{email}, null, null, null, null);</code> |
| 41 | |
| 42 | <code> while (cursor.moveToNext()) {</code> |
| 43 | <code> if (cursor.isFirst()) {</code> |
| 44 | <code> cursor.moveToFirst();</code> |
| 45 | <code> // If Email is already exists then Result variable</code> |
| 46 | <code>value set as Email Found.</code> |
| 47 | <code> F_Result = "Email Found";</code> |
| 48 | |
| 49 | <code> // Closing cursor.</code> |
| 50 | <code> cursor.close();</code> |
| 51 | <code> kirimpesan();</code> |

| | |
|----|--|
| 52 | } |
| 53 | } |
| 54 | |
| 55 | <i>// Calling method to check final result and insert data into</i> |
| 56 | <i>SQLite database.</i> |
| 57 | <code>Toast.makeText(controller_pesan.this, "Email penerima tidak</code> |
| 58 | <code>terdaftar", Toast.LENGTH_LONG).show();</code> |
| 59 | |
| 60 | } |

Pada baris ke 1 ditunjukkan bahwa fungsi mengirim pesan akan terjadi apabila *user* mengklik tombol simpan. Lalu langkah awal yang terjadi adalah memastikan email yang dituju sudah terdaftar, fungsi itu terdapat pada baris ke 30 dan di panggil pada baris 4. Apabila email terdaftar, kemudian fungsi kirim pesan dipanggil dan menyimpan konten pesan yang dimasukan *user* kedalam database.

5.10 Implementasi Fitur Membalas Pesan Ke Ahli Gizi

Implementasi fitur membalas pesan ke ahli gizi pada aplikasi perangkat bergerak ini berguna bagi *user* untuk membalas pesan ke ahli gizi, berkaitan dengan janji konsultasi yang ingin dilakukan oleh *user* nantinya. Pada tabel 5.10 dibawah ini merupakan *Source Code* implementasi membalas pesan ke ahli gizi yang terdapat pada *class* controller_tulispesan.

Tabel 5.10 Source Code Implementasi Membalas Pesan Ke Ahli Gizi

| No | Source Code |
|----|--|
| 1 | <code>kirim.setOnClickListener(new View.OnClickListener() {</code> |
| 2 | <code>@Override</code> |
| 3 | <code>public void onClick(View view) {</code> |
| 4 | <code>CheckingEmailAlreadyExistsOrNot();</code> |
| 5 | |
| 6 | |
| 7 | <code>}</code> |
| 8 | <code>});</code> |
| 9 | |
| 10 | <code>}</code> |
| 11 | <code>public void kirimpesan(){</code> |
| 12 | <code>inputancontent = holdercontent.getText().toString();</code> |
| 13 | <i>// Opening new user registration activity using intent on button</i> |
| 14 | <i>click.</i> |
| 15 | <code>SQLiteDatabase sqLiteDatabaseObj =</code> |
| 16 | <code>model_janjihelper.getReadableDatabase();</code> |
| 17 | |
| 18 | <code>model_janjihelper.onCreate(sqLiteDatabaseObj);</code> |
| 19 | |
| 20 | <code>model_janjihelper.tambahjanji(sqLiteDatabaseObj, sessionemail, email, i</code> |
| 21 | <code>nputancontent);</code> |
| 22 | <code>Toast.makeText(controller_pesan.this, "Pesan Berhasil</code> |
| 23 | <code>dikirim", Toast.LENGTH_LONG).show();</code> |
| 24 | <code>Intent intent = new Intent(controller_pesan.this,</code> |
| 25 | <code>controller_home.class);</code> |
| 26 | <code>startActivity(intent);</code> |
| 27 | |
| 28 | <code>}</code> |
| 29 | |
| 30 | <code>public void CheckingEmailAlreadyExistsOrNot() {</code> |
| 31 | |
| 32 | <i>// Opening SQLite database write permission.</i> |
| 33 | <code>sqLiteDatabaseObj2 =</code> |

```

34 model_ahligizihelper.getWritableDatabase();
35
36 // Adding search email query to cursor.
37 cursor =
38 SQLiteDatabaseObj2.query(model_ahligizihelper.TABLE_NAME, null, " "
39 + model_ahligizihelper.Table_Column_2_email + "=?", new
40 String[]{email}, null, null, null, null);
41
42 while (cursor.moveToNext()) {
43     if (cursor.isFirst()) {
44         cursor.moveToFirst();
45         // If Email is already exists then Result variable
46         value set as Email Found.
47         F_Result = "Email Found";
48
49         // Closing cursor.
50         cursor.close();
51         kirimpesan();
52     }
53 }
54
55 // Calling method to check final result and insert data into
56 SQLite database.
57 Toast.makeText(controller_pesan.this, "Email penerima tidak
58 terdaftar", Toast.LENGTH_LONG).show();
59
60 }

```

Pada baris ke 1 ditunjukkan bahwa fungsi mengirim pesan akan terjadi apabila *user* mengklik tombol simpan. Lalu langkah awal yang terjadi adalah memastikan email yang dituju sudah terdaftar, fungsi itu terdapat pada baris ke 30 dan di panggil pada baris 4. Apabila email terdaftar, kemudian fungsi kirim pesan dipanggil dan menyimpan konten pesan yang dimasukan *user* kedalam database.

5.11 Implementasi Fitur Melakukan Logout

Implementasi fitur melakukan *logout* ini berfungsi bagi *user* untuk melakukan *logout* terhadap akun miliknya pada aplikasi. Tabel 5.11 merupakan implementasi dari fitur melakukan *logout* pengguna yang terdapat pada class *controller_home*.

Tabel 5.11 Source Code Implementasi Membalas Pesan Ke Ahli Gizi

| No | Source Code |
|----|--|
| 1 | public void redirect_logout(View view){ |
| 2 | Intent redirect = new Intent(controller_home.this, |
| 3 | SplashScreen.class); |
| 4 | startActivity(redirect); |
| 5 | sessionemail == null; |
| 6 | } |

Fungsi utama dari melakukan *logout* pada class *controllerhome* adalah menghilangkan session email yang terdapat pada baris kelima pada tabel 5.11 diatas.

5.12 Implementasi Fitur Melakukan Registrasi Ahli Gizi

Implementasi ini bermanfaat bagi ahli gizi untuk melakukan registrasi terhadap aplikasi. Fitur ini diimplementasikan pada class

controller_register_ahligizi. Tabel 5.12 dibawah ini merupakan hasil dari implementasi fitur tersebut.

Tabel 5.12 Source Code Implementasi Fitur Register Ahli Gizi

| No | Source Code |
|----|--|
| 1 | <code>public void SQLiteTableBuild() {</code> |
| 2 | |
| 3 | <code>sqliteDatabaseObj.execSQL("CREATE TABLE IF NOT EXISTS " +</code> |
| 4 | <code>model_ahligizihelper.TABLE_NAME + "("</code> |
| 5 | <code>+ model_ahligizihelper.Table_Column_ID + " INTEGER</code> |
| 6 | <code>PRIMARY KEY AUTOINCREMENT NOT NULL, "</code> |
| 7 | <code>+ model_ahligizihelper.Table_Column_1_nama + "</code> |
| 8 | <code>VARCHAR, "</code> |
| 9 | <code>+ model_ahligizihelper.Table_Column_2_email + "</code> |
| 10 | <code>VARCHAR, "</code> |
| 11 | <code>+ model_ahligizihelper.Table_Column_3_password+"</code> |
| 12 | <code>VARCHAR, "</code> |
| 13 | <code>+ model_ahligizihelper.Table_Column_4_notlp+"</code> |
| 14 | <code>VARCHAR, "</code> |
| 15 | <code>+ model_ahligizihelper.Table_Column_5_biodata+"</code> |
| 16 | <code>VARCHAR);");</code> |
| 17 | <code>}</code> |
| 18 | |
| 19 | <code>public void CheckEditTextStatus(){</code> |
| 20 | <code>// Getting value from All EditText and storing into String</code> |
| 21 | <code>Variables.</code> |
| 22 | <code>inputnama = holdernama.getText().toString() ;</code> |
| 23 | <code>inputemail = holderemail.getText().toString();</code> |
| 24 | <code>inputpassword = holderpassword.getText().toString();</code> |
| 25 | <code>if(TextUtils.isEmpty(inputnama) </code> |
| 26 | <code>TextUtils.isEmpty(inputemail) TextUtils.isEmpty(inputpassword)){</code> |
| 27 | <code>EditTextEmptyHolder = false ;</code> |
| 28 | <code>Toast.makeText(controller_register_ahligizi.this,</code> |
| 29 | <code>"Tolong Lengkapi Username dan Password anda.",</code> |
| 30 | <code>Toast.LENGTH_LONG).show();</code> |
| 31 | <code>}</code> |
| 32 | <code>else {</code> |
| 33 | <code>EditTextEmptyHolder = true ;</code> |
| 34 | <code>}</code> |
| 35 | |
| 36 | |
| 37 | <code>public void CheckingEmailAlreadyExistsOrNot(){</code> |
| 38 | <code>// Opening SQLite database write permission.</code> |
| 39 | <code>sqliteDatabaseObj =</code> |
| 40 | <code>model_ahligizihelper.getWritableDatabase();</code> |
| 41 | <code>// Adding search email query to cursor.</code> |
| 42 | <code>cursor =</code> |
| 43 | <code>sqliteDatabaseObj.query(model_ahligizihelper.TABLE_NAME, null, " "</code> |
| 44 | <code>+ model_ahligizihelper.Table_Column_2_email + "=?", new</code> |
| 45 | <code>String[]{inputemail}, null, null, null, null);</code> |
| 46 | |
| 47 | <code>while (cursor.moveToNext()) {</code> |
| 48 | <code>if (cursor.isFirst()) {</code> |
| 49 | <code>cursor.moveToFirst();</code> |
| 50 | <code>// If Email is already exists then Result variable</code> |
| 51 | <code>value set as Email Found.</code> |
| 52 | <code>F_Result = "Email Found";</code> |
| 53 | <code>// Closing cursor.</code> |
| 54 | <code>cursor.close();</code> |
| 55 | <code>}</code> |
| 56 | <code>}</code> |
| 57 | <code>// Calling method to check final result and insert data</code> |
| 58 | <code>into SQLite database.</code> |


```

59         CheckFinalResult();
60     }
61     public void CheckFinalResult(){
62         // Checking whether email is already exists or not.
63         if(F_Result.equalsIgnoreCase("Email Found"))
64         {
65             // If email is exists then toast msg will display.
66             //
67             model_userhelper.deleteuser(sqliteDatabaseObj,inputnama);
68             Toast.makeText(controller_register_ahligizi.this,"Email
69 Already Exists",Toast.LENGTH_LONG).show();
70         }
71         else{
72             // If email already dose n't exists then user
73             registration details will entered to SQLite database.
74             if(EditTextEmptyHolder == true)
75             {
76                 // SQLite query to insert data into table.
77
78             model_ahligizihelper.tambahahligizi(sqliteDatabaseObj,inputnama,inp
79 utemail,inputpassword);
80                 //pemberitahuan register berhasil
81
82             Toast.makeText(controller_register_ahligizi.this,"Registrasi
83 berhasil.",Toast.LENGTH_LONG).show();
84             Intent intent = new
85             Intent(controller_register_ahligizi.this,controller_biodata_ahligiz
86 i.class);
87                 intent.putExtra("UserEmail", inputemail);
88                 startActivity(intent);
89             }
90             // This block will execute if any of the registration
91             EditText is empty.
92             else {
93                 // pemberitahuan inputan kosong
94
95             Toast.makeText(controller_register_ahligizi.this,"Data anda masih
96 kosong.", Toast.LENGTH_LONG).show();
97             }
98         }
99         F_Result = "Not_Found" ;
100     }

```

Baris ke 1 sampai 16 merupakan fungsi untuk melakukan hubungan dengan database ahli gizi yang di modelkan dalam class model_ahligizi. Selanjutnya pada baris ke 19 sampai dengan 34 merupakan fungsi untuk melakukan pengecekan apakah ada form yang tidak diisi atau kosong.

Lalu isian dari form tersebut dicek apakah email yang ingin digunakan sudah pernah digunakan sebelumnya pada baris ke 37. Apabila email yang ingin digunakan tidak pernah dipakai oleh orang lain barulah fungsi untuk melakukan registrasi dilakukan dengan memanggil fungsi tambahahligizpada class model_ahligizi yang dapat dilihat pada baris ke 78.

5.13 Implementasi Fitur Melakukan *Login* Ahli Gizi

Implementasi fitur *login* pada aplikasi perangkat bergerak ini berguna untuk melakukan autentikasi *user* sehingga data yang diberikan aplikasi ini nantinya sesuai dengan biodata *user* yang telah diberikan saat melakukan registrasi. Implementasi fitur *login* pada aplikasi ini dilakukan pada *class* *controller_login_ahligizi*. Tabel 5.2 dibawah ini merupakan *Source Code* dari implementasi fitur *login* pada *class* *controller_login_ahligizi*.

Tabel 5.13 Source Code Implementasi Fitur Login

| No | Source Code |
|----|---|
| 1 | <code>public void LoginFunction() {</code> |
| 2 | |
| 3 | <code> if (EditTextEmptyHolder) {</code> |
| 4 | |
| 5 | <code> // Opening SQLite database write permission.</code> |
| 6 | <code> sqliteDatabaseObj = model_userhelper.getWritableDatabase();</code> |
| 7 | |
| 8 | <code> // Adding search email query to cursor.</code> |
| 9 | <code> cursor =</code> |
| 10 | <code> sqliteDatabaseObj.query(model_ahligizihelper.TABLE_NAME, null, " " +</code> |
| 11 | <code> model_ahligizihelper.Table_Column_2_email + "=?", new</code> |
| 12 | <code> String[]{inputanemail}, null, null, null);</code> |
| 13 | |
| 14 | <code> while (cursor.moveToNext()) {</code> |
| 15 | |
| 16 | <code> if (cursor.isFirst()) {</code> |
| 17 | |
| 18 | <code> cursor.moveToFirst();</code> |
| 19 | |
| 20 | <code> // Storing Password associated with entered email.</code> |
| 21 | <code> TempPassword =</code> |
| 22 | <code> cursor.getString(cursor.getColumnIndex(model_ahligizihelper.Table_Co</code> |
| 23 | <code> lumn_3_password));</code> |
| 24 | |
| 25 | <code> // Closing cursor.</code> |
| 26 | <code> cursor.close();</code> |
| 27 | <code> }</code> |
| 28 | <code> }</code> |
| 29 | |
| 30 | <code> // Calling method to check final result ..</code> |
| 31 | <code> CheckFinalResult();</code> |
| 32 | |
| 33 | <code> } else {</code> |
| 34 | |
| 35 | <code> //If any of login EditText empty then this block will be</code> |
| 36 | <code> executed.</code> |
| 37 | <code> Toast.makeText(controller_login.this, "Tolong masukan email</code> |
| 38 | <code> dan password yang tepat.", Toast.LENGTH_LONG).show();</code> |
| 39 | |
| 40 | <code> }</code> |
| 41 | |
| 42 | <code>}</code> |
| 43 | |
| 44 | <code>public void CheckFinalResult() {</code> |
| 45 | |
| 46 | <code> if (TempPassword.equalsIgnoreCase(inputanpassword)) {</code> |
| 47 | |
| 48 | <code> Toast.makeText(controller_login.this, "Login Successfully",</code> |
| 49 | <code> Toast.LENGTH_LONG).show();</code> |
| 50 | <code> session = getSharedPreferences(MyPREFERENCES,</code> |

```

51 Context.MODE_PRIVATE);
52     SharedPreferences.Editor editor = session.edit();
53     editor.putString("EmailSession", inputanemail);
54     editor.commit();
55
56     sessionemail();
57     // Going to Dashboard activity after login success message.
58     Intent intent = new Intent(controller_login.this,
59 controller_home.class);
60     startActivity(intent);
61
62 } else {
63     cobal.setText(TempPassword);
64     Toast.makeText(controller_login.this, "UserName or Password
65 is Wrong, Please Try Again.", Toast.LENGTH_LONG).show();
66
67 }
68 }

```

Baris ke 9 sampai dengan 12 merupakan *query* untuk melakukan pengecekan apakah email yang dimasukan *user* terdaftar pada database atau tidak. Lalu dari pengecekan email tersebut dilakukan pengecekan password pada fungsi *CheckFinalResult* yang dipanggil pada baris 31.

Pada baris ke 44 dilakukan pengecekan password dari *user* dengan password yang terdaftar pada database. Apabila password tersebut benar maka *user* akan langsung diarahkan ke halaman beranda.

5.14 Implementasi Fitur Menambahkan Biodata Ahli Gizi

Implementasi fitur masukan biodata pada aplikasi perangkat bergerak ini berguna untuk menerima biodata *user* seperti tinggi badan, berat badan dan tanggal lahir. Dimana biodata tersebut bermanfaat sebagai parameter untuk melakukan perhitungan kalori menggunakan metode *harrise bendict* nantinya. Implementasi fitur masukan biodata pada aplikasi perangkat bergerak ini dilakukan pada *class* *controller_biodata_ahligizi*. Tabel 5.3 dibawah ini merupakan *source code* dari implementasi fitur masukan biodata.

Tabel 5.14 Source Code Implementasi Fitur Tambah Biodata

| No | Source Code |
|----|---|
| 1 | public void tambahbiodata() { |
| 2 | SQLiteDatabaseBuild(); |
| 3 | int tinggibadan = |
| 4 | Integer.parseInt(holdertinggibadan.getText().toString()); |
| 5 | int beratbadan = |
| 6 | Integer.parseInt(holderberatbadan.getText().toString()); |
| 7 | |
| 8 | sqLiteDatabaseObj = model_userhelper.getWritableDatabase(); |
| 9 | model_ahligizihelper.tambahbiodataahligizi |
| 10 | (sqLiteDatabaseObj,email,biodata,notelp,linkfoto); |
| 11 | |
| 12 | Intent intent = new Intent(controller_biodata.this, |
| 13 | controller_login.class); |
| 14 | startActivity(intent); |
| 15 | } |

Pada baris 4 sampai 6 dilakukan pengambilan biodata dan no telp dari ahli gizi. Kemudian dilakukan pemanggilan fungsi tambahbiodataahligizi yang terdapat pada *class* model_ahligizi dengan.

5.15 Implementasi Fitur Memperbarui Biodata Ahli Gizi

Implementasi fitur memperbarui biodata pada aplikasi perangkat bergerak ini berguna untuk melakukan pemberauman biodata pada aplikasi, apabila seorang ahli gizi ingin memperbarui biodata pada aplikasi. Pada tabel 5.15 merupakan implementasi *source code* dari fitur memperbarui biodata.

Tabel 5.15 Source Code Implementasi Fitur Memperbarui Biodata

| No | Source Code |
|----|--|
| 1 | public void tambahbiodata(SQLiteDatabase sqLiteDatabaseObj, String |
| 2 | email,String biodata, String notelp){ |
| 3 | SQLiteDatabaseBuild(); |
| 4 | sqLiteDatabaseObj =model_ahligizihelper.getWritableDatabase(); |
| 5 | model_ahligizihelper.tambahbiodataahligizi(sqLiteDatabaseObj,email, |
| 6 | biodata,notelp,linkfoto); |
| 7 | Intent intent = new Intent(controller_biodata_ahligizi. this , |
| 8 | controller_login_ahligizi. class); |
| 9 | startActivity(intent); |
| 10 | } |

Baris pertama sampai terakhir pada tabel 5.15 diatas merupakan fungsi dari menambahkan biodata pada database ahligizi. Dapat dilihat pada baris ke 4 fungsi tersebut memanggil fungsi tambahbiodataahligizi yang terdapat pada class model_ahligizi.

5.16 Implementasi Fitur Membalas Pesan Ke Pengguna

Implementasi fitur membalas pesan ke ahli gizi pada aplikasi perangkat bergerak ini berguna bagi ahli gizi untuk membalas pesan ke pengguna, berkaitan dengan janji konsultasi yang ingin dilakukan oleh *pengguna* nantinya. Pada tabel 5.16 dibawah ini merupakan *Source Code* implementasi membalas pesan ke ahli pengguna yang terdapat pada *class* controller_tulispesan_ahligizi.

Tabel 5.16 Source Code Implementasi Membalas Pesan Ke Ahli Gizi

| No | Source Code |
|----|---|
| 1 | public void kirimpesan(){ |
| 2 | <inputancontent> = holdercontent.getText().toString();</inputancontent> |
| 3 | // Opening new user registration activity using intent on |
| 4 | button click. |
| 5 | SQLiteDatabase sqLiteDatabaseObj = |
| 6 | model_janjihelper.getReadableDatabase(); |
| 7 | |
| 8 | model_janjihelper.onCreate(sqLiteDatabaseObj); |
| 9 | |
| 10 | model_janjihelper.tambahpesan(sqLiteDatabaseObj,sessionemail,email,i |
| 11 | nputancontent); |
| 12 | Toast.makeText(controller_tulis_pesan_ahligizi. this , "Pesan |
| 13 | Berhasil dikirim",Toast.LENGTH_LONG).show(); |
| 14 | Intent intent = new |
| 15 | Intent(controller_tulis_pesan_ahligizi. this , |
| 16 | controller_home_ahligizi. class); |
| 17 | startActivity(intent); |
| 18 | |

```

19     }
20
21     public void CheckingEmailAlreadyExistsOrNot(){
22         // Opening SQLite database write permission.
23         SQLiteDatabaseObj2 = model_userhelper.getWritableDatabase();
24         // Adding search email query to cursor.
25         cursor =
26         SQLiteDatabaseObj2.query(model_userhelper.TABLE_NAME, null, " " +
27         model_userhelper.Table_Column_2_email + "=?", new String[]{email},
28         null, null, null, null);
29
30         while (cursor.moveToNext()) {
31             if (cursor.isFirst()) {
32                 cursor.moveToFirst();
33                 // If Email is already exists then Result variable
34                 value set as Email Found.
35                 F_Result = "Email Found";
36
37                 // Closing cursor.
38                 cursor.close();
39                 kirimpesan();
40             }
41             if (F_Result == "Not Found") {
42                 Toast.makeText(controller_tulis_pesan_ahligizi.this,
43                 "Email penerima tidak terdaftar", Toast.LENGTH_LONG).show();
44             }
45         }
46         // Calling method to check final result and insert data into
47         SQLite database.
48     }
49 }
50
51
52
53
54

```

Pada baris ke 1 ditunjukkan bahwa fungsi mengirim pesan akan terjadi apabila ahli gizi mengklik tombol simpan. Lalu langkah awal yang terjadi adalah memastikan email yang dituju sudah terdaftar, fungsi itu terdapat pada baris ke 21. Apabila email terdaftar, kemudian fungsi kirim pesan dipanggil dan menyimpan konten pesan yang dimasukan ahli gizi kedalam database dimana fungsi tersebut dipanggil pada baris ke 39.

5.17 Implementasi Fitur Menghitung Kebutuhan Kalori perhari

Implementasi fitur hitung kalori pada aplikasi perangkat bergerak ini berguna untuk menentukan jumlah kalori yang dibutuhkan *user* menggunakan metode *harris benedict*. Pada tabel 5.17 dibawa ini merupakan *Source Code* implementasi fitur hitung kalori yang terdapat pada *class* *controller_hitungkalori_ahligizi*.

Tabel 5.17 Source Code Implementasi Fitur Hitung Kalori

| No | Source Code |
|----|--|
| 1 | public void hitungkalori (View view){ |
| 2 | nilaiberatbadan = |
| 3 | Integer.parseInt(beratbadan .getText().toString()); |
| 4 | nilaitinggibadan = |
| 5 | Integer.parseInt(tinggibadan .getText().toString()); |
| 6 | nilaiumur = Integer.parseInt(umur .getText().toString()); |

```

7 //metodeharrisbenedict
8 double AMB = 655+ (9.6 * nilaiberatbadan) + (1.8 *
9 nilaitinggibadan) - (4.7 * nilaiumur);
10 double energi = AMB * nilaistress * nilaiaktivitas;
11 hasil = String.valueOf(energi);
12 hitung.setText(hasil);
13 int i= (int) Math.round(energi);
14 model_userhelper.tambahkaloriuser(sqliteDatabaseObj2
15 ,i,sessionemail);
16 Intent intent = new Intent(controller_hitungkalori.this,
17 controller_profil.class);
18 startActivity(intent);
19 }

```

Pada baris ke 3 sampai ke 6 merupakan pengambilan data berat badan, tinggi badan dan umur dari inputan *user*. Lalu pada baris 8 dimulailah penghitungan kalori menggunakan

5.18 Implementasi Fitur Melakukan *Logout* Ahli Gizi

Implementasi fitur melakukan *logout* ini berfungsi bagi *user* untuk melakukan *logout* terhadap akun miliknya pada aplikasi. Tabel 5.18 merupakan implementasi dari fitur melakukan *logout* ahli gizi yang terdapat pada class *controller_home*.

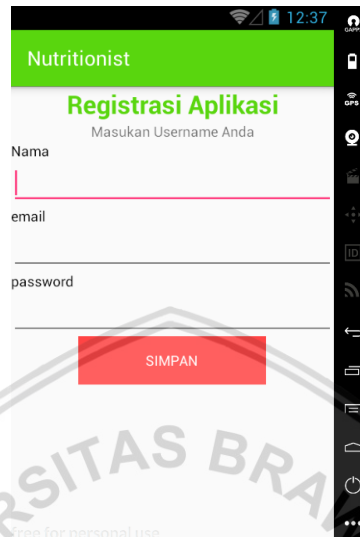
Tabel 5.18 Source Code Implementasi Membalas Pesan Ke Ahli Gizi

| No | Source Code |
|----|--|
| 1 | public void redirect_logout(View view){ |
| 2 | Intent redirect = new Intent(controller_home.this, |
| 3 | SplashScreen.class); |
| 4 | startActivity(redirect); |
| 5 | sessionemail == null; |
| 6 | } |

Fungsi utama dari melakukan *logout* pada class *controller_home_ahligizi* adalah menghilangkan session email yang terdapat pada baris kelima pada tabel 5.18 diatas.

5.19 Implementasi *User Interface Register*

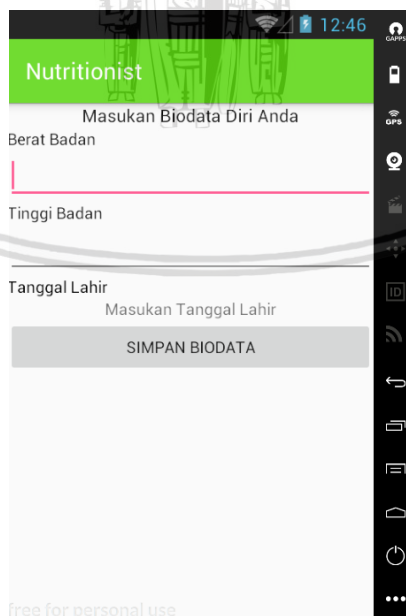
Implementasi *user interface register* pada aplikasi perangkat bergerak ini dilakukan pada *class view_register*. gambar 5.1 merupakan hasil dari *interface login* dari aplikasi ini .



Gambar 5.1 *User Interface Registrasi*

5.20 Implementasi *User Interface Masukan Biodata*

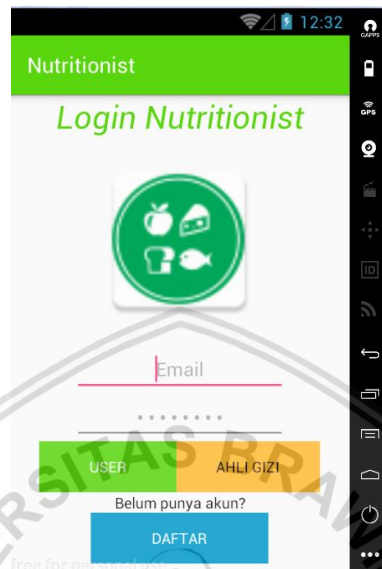
Implementasi *user interface masukan biodata* pada aplikasi perangkat bergerak ini dilakukan pada *class view_biodata*. gambar 5.2 merupakan hasil dari *interface masukan biodata* dari aplikasi ini .



Gambar 5.2 *User Interface Masukan Biodata*

5.21 Implementasi *User Interface Login*

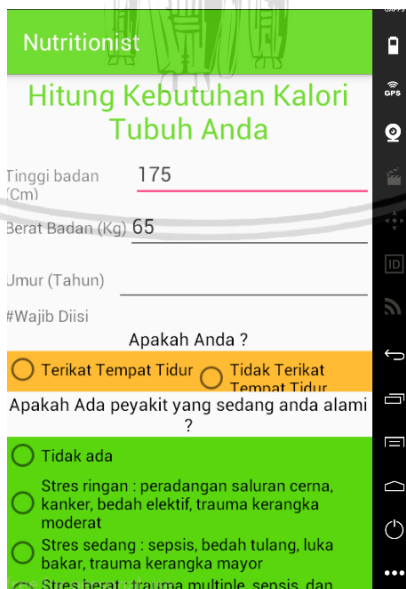
Implementasi *user interface login* pada aplikasi perangkat bergerak ini dilakukan pada *class view_login*. Gambar 5.2 merupakan hasil dari *interface login* dari aplikasi ini .



Gambar 5.3 Gambar *User Interface Login*

5.22 Implementasi *User Interface Hitung Kalori*

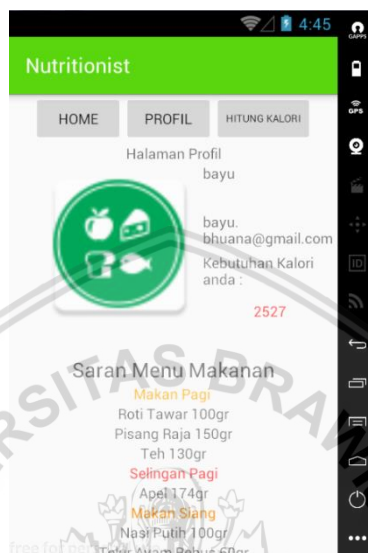
Implementasi *user interface* hitung kalori pada aplikasi perangkat bergerak ini dilakukan pada *class view_hitungkalori*. Gambar 5.3 merupakan hasil dari implementasi *user interface* hitung kalori.



Gambar 5.4 Gambar *User Interface Hitung Kalori*

5.23 Implementasi *User Interface* Menampilkan Saran Menu Makanan

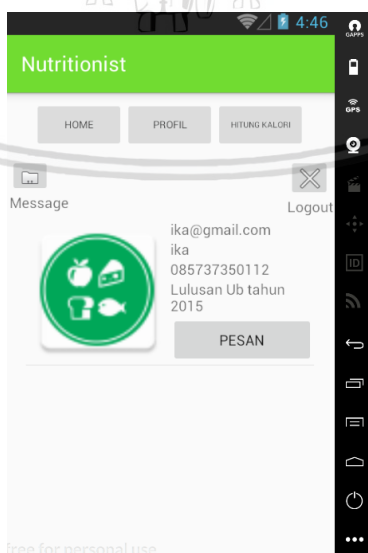
Implementasi *user interface* menampilkan saran menu makanan pada aplikasi perangkat bergerak ini dilakukan pada *class view_profil*. Gambar 5.5 merupakan hasil dari *interface* menampilkan saran menu makanan dari aplikasi ini .



Gambar 5.5 *User Interface* Menampilkan Saran Menu Makanan

5.24 Implementasi *User Interface* Menampilkan *List* Ahli Gizi

Implementasi *user interface* menampilkan *List* ahli gizi pada aplikasi perangkat bergerak ini dilakukan pada *class view_home*. Gambar 5.6 merupakan hasil dari *interface* menampilkan *List* ahli gizi dari aplikasi ini .



Gambar 5.6 *User Interface* Menampilkan *List* Ahli Gizi

5.25 Implementasi *User Interface* Mengirim Pesan ke Ahli Gizi

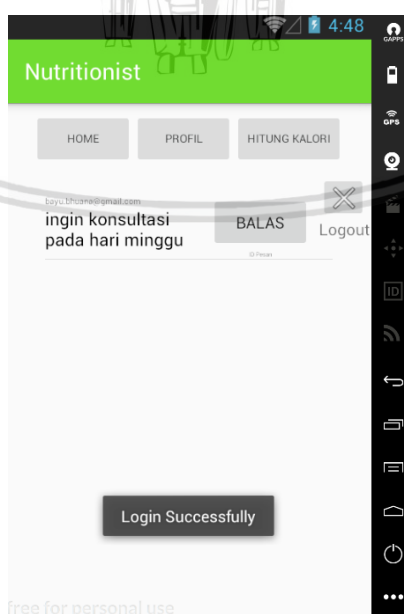
Implementasi *user interface* mengirim pesan ke ahli gizi pada aplikasi perangkat bergerak ini dilakukan pada *class view_pesanan*. Gambar 5.7 merupakan hasil dari *interface* menampilkan *List* ahli gizi dari aplikasi ini .



Gambar 5.7 *User Interface* Mengirim Pesan ke Ahli Gizi

5.26 Implementasi *User Interface* Melihat Pesan

Implementasi *user interface* menampilkan saran menu makanan pada aplikasi perangkat bergerak ini dilakukan pada *class view_janji*. Gambar 5.8 merupakan hasil dari *interface* melihat pesan dari aplikasi ini .



Gambar 5.8 *User Interface* Melihat Pesan

5.27 Implementasi *User Interface* Membalas Pesan

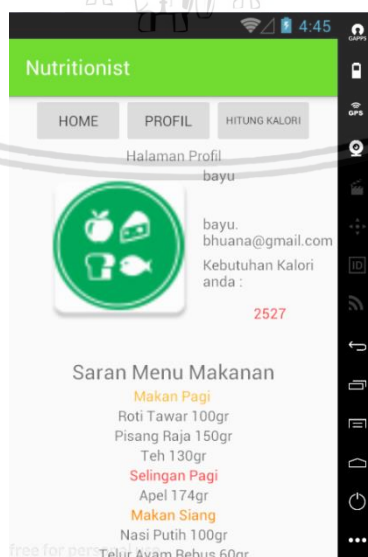
Implementasi *user interface* membalas pesan pada aplikasi perangkat bergerak ini dilakukan pada *class view_pesanan*. Gambar 5.9 merupakan hasil dari *interface* menampilkan saran menu makanan dari aplikasi ini .



Gambar 5.9 *User Inter Face* Membalas Pesan

5.28 Implementasi *User Interface* Melihat Biodata User

Implementasi *user interface* menampilkan saran menu makanan pada aplikasi perangkat bergerak ini dilakukan pada *class view_biodatauser*. Gambar 5.10 merupakan hasil dari *interface* menampilkan saran menu makanan dari aplikasi ini .



Gambar 5.10 *User Interface* Melihat biodata User

BAB 6 PENGUJIAN DAN ANALISIS

Bab ini membahas mengenai tahapan pengujian yang dilakukan pada aplikasi konsultasi menu makanan bagi penderita penyakit diabetes menggunakan metode *harris benedict*. Pengujian yang dilakukan yaitu pengujian unit (*whitebox*), pengujian validasi dan pengujian usabilitas.

6.1 Pengujian Unit

Pengujian unit merupakan sebuah pengujian yang dilakukan pada unit - unit yang ada diimplementasikan di aplikasi perangkat bergerak ini. Hasil yang didapat dari pengujian ini merupakan nilai kompleksitas algoritma yang terdapat didalam sebuah unit lalu menggunakan nilai itu untuk *basis set*. Berikut merupakan Tahapan – tahapan yang dilakukan dalam pengujian unit ini.

1. Pembuatan *diagram* alir dari algoritma *pseudocode*
2. Menentukan *cyclomatic complexity* dari *diagram* alir
3. Menentukan *Independen Path*

6.1.1 Pengujian Unit Fitur Menampilkan Menu Makanan

1. *Pseudocode*

Tabel 6.1 merupakan *pseudocode* dari fungsi untuk menampilkan menu makanan yang ada pada aplikasi ini.

Tabel 6.1 Algoritma Fitur Menampilkan Menu Makanan

| no | <i>pseudocode</i> |
|----|--|
| 1 | public void menampilkan_menumakanan(int kalori, String jk) { |
| 2 | //jika jenis kelamin laki laki |
| 3 | int kalori = kalori |
| 4 | string jk = jk |
| 5 | if ((kalori >= 2725) && (jk.equals("L"))) { |
| 6 | makanpagi1.setText("Bubur Tawar 200gr"); |
| 7 | makanpagi2.setText("Sop Bayam 150gr"); |
| 8 | makanpagi3.setText("Telur Dadar 40gr"); |
| 9 | makanpagi4.setText("Apel Merah 140gr"); |
| 10 | makanpagi5.setText("Teh 140 ml"); |
| 11 | |
| 12 | selinganpagi1.setText("Pisang 125gr"); |
| 13 | |
| 14 | makansiang1.setText("Nasi Putih 100gr"); |
| 15 | makansiang2.setText("Telur Ayam Rebus 60gr"); |
| 16 | makansiang3.setText("Sop Mutiara Jagung 200gr"); |
| 17 | makansiang4.setText("Apel 320 gr"); |
| 18 | makansiang5.setText("Pir 440 gr"); |
| 19 | |
| 20 | selingan sore1.setText("Kue Ape 210gr"); |
| 21 | |
| 22 | makanmalam1.setText("Nasi Putih 100gr"); |
| 23 | makanmalam2.setText("Ayam Rebus 100gr"); |
| 24 | makanmalam3.setText("Melon 200 gr"); |
| 25 | makanmalam4.setText("Teh Cangkir 140 ml"); |
| 26 | |
| 27 | selinganmalam1.setText("Susu 100gr"); |
| 28 | selinganmalam2.setText("Roti Tawar 100gr"); |
| 29 | |
| 30 | if (kalori > 2725) { |


```

31         saran.setText("Anda harus melakukan konsultasi");
32         dengan ahli gizi untuk menu makanan yang lebih cocok");
33     }
34 }
35 //menumakanan kalori kurangdari 2725 lebih dari 2625
36 if ((kalori > 2625) && (kalori < 2725)&& (jk.equals("L"))))
37 {
38     makanpagi1.setText("jagung Rebus 312 gr");
39     makanpagi2.setText("Pisang Hijau 200 gr");
40     makanpagi3.setText("Telur Dadar 40gr");
41     makanpagi4.setText("Apel 140gr");
42     makanpagi5.setText("Teh Cangkir 150 ml");
43
44     selinganpagi1.setText("Pir 440gr");
45     selinganpagi1.setText("Teh Tawar 100 ml");
46
47     makansiang1.setText("Nasi Putih 100gr");
48     makansiang2.setText("Telur Ayam Rebus 60gr");
49     makansiang3.setText("Ikan Teri 50 gr");
50     makansiang4.setText("Tumis Buncis 100 gr");
51     makansiang5.setText("Pir 440 gr");
52     makansiang6.setText("Jambu Air 60 gr");
53     makansiang7.setText("Jus Melon 150 ml");
54
55     selingansore1.setText("Apel Merah 140gr");
56     selingansore2.setText("Pepaya 100gr");
57     selingansore3.setText("Jambu Biji 320gr");
58
59     makanmalaml1.setText("Nasi Putih 100gr");
60     makanmalaml2.setText("Ayam Rebus 100gr");
61     makanmalaml3.setText("tumis Buncis 200 gr");
62     makanmalaml4.setText("Teh Tawar 100 ml");
63
64     selinganmalaml1.setText("Susu 100gr");
65     selinganmalaml2.setText("Roti Tawar 50gr");
66     selinganmalaml3.setText("Pepaya 200 gr");
67 }
68
69
70 if ((kalori <= 2625) && (jk.equals("L")))) {
71     makanpagi1.setText("Bubur 300 gr");
72     makanpagi2.setText("Telur Dadar 80gr");
73     makanpagi3.setText("Apel 160gr");
74     makanpagi4.setText("Teh Tawar 100 ml");
75
76     selinganpagi1.setText("Cracker 50 gr");
77     selinganpagi2.setText("Teh Tawar 100 ml");
78
79     makansiang1.setText("Nasi Putih 100gr");
80     makansiang2.setText("Telur Ayam Rebus 60gr");
81     makansiang3.setText("Ikan Teri 50 gr");
82     makansiang4.setText("Sayur Lodeh 100 gr");
83     makansiang5.setText("Pir 440 gr");
84     makansiang6.setText("Jambu Air 60 gr");
85     makansiang7.setText("Jus Melon 150 ml");
86
87     selingansore1.setText("Apel Merah 70gr");
88     selingansore2.setText("Jambu Biji 320gr");
89
90     makanmalaml1.setText("Nasi Putih 100gr");
91     makanmalaml2.setText("Ayam Rebus 100gr");
92     makanmalaml3.setText("tumis Buncis 100 gr");
93     makanmalaml4.setText("Teh Tawar 100 ml");
94
95     selinganmalaml1.setText("Susu 100gr");

```

```

96         selinganmalam2.setText("Roti Tawar 50gr");
97         selinganmalam3.setText("Pepaya 200 gr");
98
99         if (kalori < 2625) { 10
100             saran.setText("Anda harus melakukan konsulta 11
101 dengan ahli gizi untuk menu makanan yang lebih cocok");
102         }
103     }
104     if ((kalori >= 2250) && ((jk.equals("P")))) { 12
105         makanpagi1.setText("Bubur ayam 200gr");
106         makanpagi2.setText("Telur Dadar 80gr");
107         makanpagi3.setText("Teh Tawar 140 ml");
108
109         selinganpagi1.setText("Cracker 50 gr");
110
111         makansiang1.setText("Nasi Putih 100gr");
112         makansiang2.setText("Telur Ayam Rebus 100gr");
113         makansiang3.setText("Sayur Lodeh 100gr");
114         makansiang4.setText("Pir 440 gr");
115         makansiang5.setText("Jus Melon 150 gr"); 13
116
117         selingansore1.setText("Cracker 50 gr");
118
119         makanmalam1.setText("Nasi Putih 100gr");
120         makanmalam2.setText("Ayam Rebus 100gr");
121         makanmalam3.setText("Tumis Buncis 50 gr");
122         makanmalam4.setText("Teh Tawar 100 ml");
123
124         selinganmalam1.setText("Roti Tawar 50gr");
125         selinganmalam2.setText("Pepaya 200gr");
126
127         if (kalori > 2250) { 14
128             saran.setText("Anda harus melakukan konsulta 15
129 dengan ahli gizi untuk menu makanan yang lebih cocok");
130         }
131     }
132     //menumakanan kalori kurangdari 2725 lebih dari 2625
133     if ((kalori > 2125) && (kalori < 2250)&& 16
134         ((jk.equals("P")))) {
135         makanpagi1.setText("Bubur Ayam 200 gr");
136         makanpagi2.setText("Roti Tawar 50 gr");
137         makanpagi3.setText("Apel 160gr");
138         makanpagi4.setText("Jus Melon 170 ml");
139
140         selinganpagi1.setText("Pisang Rebus 125gr");
141         selinganpagi1.setText("Apel 160 ml");
142
143         makansiang1.setText("Nasi Putih 100gr");
144         makansiang2.setText("Sayur Asam 120 gr");
145         makansiang3.setText("Ikan Teri 50 gr");
146         makansiang4.setText("Perkedel Jagung 50 gr");
147         makansiang5.setText("Jus Melon 150 ml");
148
149         selingansore1.setText("Cracker 50gr");
150
151         makanmalam1.setText("Nasi Putih 70 gr");
152         makanmalam2.setText("Ayam Rebus 100gr");
153         makanmalam3.setText("Teh Tawar 100 ml");
154
155         selinganmalam1.setText("Cracker 50gr");
156     }
157
158     if (kalori <= 2125 && ((jk.equals("P")))) { 18
159         makanpagi1.setText("Roti Tawar 50 gr");
160     }

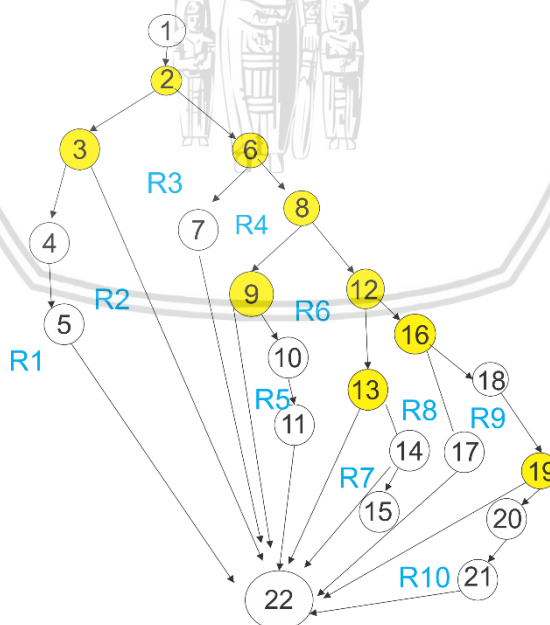
```

| | | |
|-----|---|----|
| 161 | makanpagi3.setText("Telur Dadar 75 gr"); | |
| 162 | makanpagi4.setText("Apel 160gr"); | |
| 163 | makanpagi5.setText("Jus Melon 170 ml"); | |
| 164 | | |
| 165 | selinganpagi1.setText("Pisang Rebus 125 gr"); | |
| 166 | selinganpagi1.setText("Salak 150 ml"); | |
| 167 | | |
| 168 | makansiang1.setText("Nasi Putih 100gr"); | |
| 169 | makansiang2.setText("Ikan Teri 50 gr"); | |
| 170 | makansiang3.setText("Perkedel Jagung 50 gr "); | 19 |
| 171 | makansiang4.setText("Sop Bayam 50 gr"); | |
| 172 | makansiang5.setText("Jambu Air 60 gr"); | |
| 173 | makansiang6.setText("Jus Melon 150 ml"); | |
| 174 | | |
| 175 | selingansore1.setText("Jambu Biji 320gr"); | |
| 176 | selingansore2.setText("jeruk Pontianak 145"); | |
| 177 | | |
| 178 | makanmalam1.setText("Nasi Putih 100gr"); | |
| 179 | makanmalam2.setText("Ayam Rebus 100gr"); | |
| 180 | makanmalam3.setText("tumis Buncis 100 gr"); | |
| 181 | | |
| 182 | selinganmalam1.setText("Cracker 45gr"); | |
| 183 | | |
| 184 | if (kalori < 2125 && jeniskelamin == "P") { | 20 |
| 185 | saran.setText("Anda harus melakukan konsultasi | |
| 186 | dengan ahli gizi untuk menu makanan yang lebih cocok"); | 21 |
| 187 | } | |
| | end if} | 22 |

2. Basis Path Testing

a. Flow Graph

Gambar 6.1 dibawah ini merupakan hasil gambaran dari *pseudocode* pada tabel 6.1 yang telah dibuat sebelumnya.



Gambar 6.1 Flow Graph Fitur Menampilkan Makanan

b. Cyclomatic Complexity

Dari gambar 6.1 diatas dapat kita tentukan berapa nilai $V(G)$ dengan ketentuan rumus sebagai berikut:

$$V(G) = \text{Edge} + \text{Nodes} + 2$$

$$V(G) = \text{Region}$$

$$V(G) = \text{Predicate note} + 1$$

Dari rumus diatas didapat lah hasil sebagai berikut berdasarkan gambar 6.1 *flow graph* menampilkan menu makanan yang telah dibuat

$$V(G) = 30 \text{ edges} - 22 \text{ nodes} + 2 = 10$$

$$V(G) = 10, \text{ ada } 10 \text{ region}$$

$$V(G) = 9 \text{ predicate nodes} + 1 = 10$$

c. Independen Path

$$\text{Jalur 1} = 1-2-3-4-5-22$$

$$\text{Jalur 2} = 1-2-3-22$$

$$\text{Jalur 3} = 1-2-6-7-22$$

$$\text{Jalur 4} = 1-2-6-8-9-22$$

$$\text{Jalur 5} = 1-2-6-8-9-10-11-22$$

$$\text{Jalur 6} = 1-2-6-8-12-13-22$$

$$\text{Jalur 7} = 1-2-6-8-12-13-14-15-12$$

$$\text{Jalur 8} = 1-2-6-8-12-16-17-22$$

$$\text{Jalur 9} = 1-2-6-8-12-16-18-19-22$$

$$\text{Jalur 10} = 1-2-6-8-12-16-18-19-20-22$$

3. Hasil Pengujian Unit Fitur Menampilkan Menu Makanan

Tabel 6.2 Hasil Pengujian Unit Menampilkan Menu Makanan

| No Jalur | Prosedur Jalur | Expected Result | Result | Status |
|----------|---|--|--|--------|
| 1 | Memanggil fungsi lihat menu makanan bagi seorang pria dengan kebutuhan kalori lebih dari atau sama dengan | Berhasil menampilkan menu makanan sesuai dengan kebutuhan kalori lebih dari atau sama dengan 2725 kalori bagi pria | menampilkan menu makanan yang sesuai dengan kebutuhan kalori lebih dari atau sama dengan 2725 kalori bagi pria | Valid |

| | | | | |
|---|--|---|--|-------|
| | 2725 kalori | | | |
| 2 | Memanggil fungsi lihat menu makanan bagi seorang pria dengan kebutuhan kalori lebih dari 2725 kalori | Berhasil menampilkan menu makanan sesuai dengan kebutuhan kalori lebih dari 2725 kalori bagi pria beserta pemberitahuan untuk menghubungi ahli gizi untuk menu makanan yang lebih tepat | menampilkan menu makanan yang sesuai dengan kebutuhan kalori lebih dari atau sama dengan 2725 kalori bagi pria beserta pemberitahuan untuk menghubungi ahli gizi untuk menu makanan yang lebih tepat | Valid |
| 3 | Memanggil fungsi lihat menu makanan bagi seorang pria dengan kebutuhan kalori diantara 2625 dan 2725 kalori | Berhasil menampilkan menu makanan sesuai dengan kebutuhan kalori diantara 2625 dan 2725 kalori bagi pria | menampilkan menu makanan yang sesuai dengan kebutuhan kalori diantara 2625 dan 2725 kalori bagi pria | Valid |
| 4 | Memanggil fungsi lihat menu makanan bagi seorang pria dengan kebutuhan kalori kurang dari atau sama dengan 2625 kalori | Berhasil menampilkan menu makanan sesuai dengan kebutuhan kalori kurang dari atau sama dengan 2625 kalori bagi pria | menampilkan menu makanan yang sesuai dengan kebutuhan kalori kurang dari atau sama dengan 2625 kalori bagi pria | Valid |
| 5 | Memanggil fungsi lihat menu makanan bagi seorang pria dengan kebutuhan | Berhasil menampilkan menu makanan sesuai dengan kebutuhan kalori kurang dari 2625 kalori | menampilkan menu makanan yang sesuai dengan kebutuhan kalori kurang dari atau sama dengan | Valid |

| | | | | |
|---|---|---|--|-------|
| | kalori kurang dari 2625 kalori | bagi pria beserta pemberitahuan untuk menghubungi ahli gizi untuk menu makanan yang lebih tepat | 2625 kalori bagi pria beserta pemberitahuan untuk menghubungi ahli gizi untuk menu makanan yang lebih tepat | |
| 6 | Memanggil fungsi lihat menu makanan bagi seorang wanita dengan kebutuhan kalori lebih dari atau sama dengan 2550 kalori | Berhasil menampilkan menu makanan sesuai dengan kebutuhan kalori lebih dari atau sama dengan 2550 kalori bagi wanita | menampilkan menu makanan yang sesuai dengan kebutuhan kalori lebih dari atau sama dengan 2550 kalori bagi wanita | Valid |
| 7 | Memanggil fungsi lihat menu makanan bagi seorang wanita dengan kebutuhan kalori lebih dari 2550 kalori | Berhasil menampilkan menu makanan sesuai dengan kebutuhan kalori lebih dari 2550 kalori bagi wanita beserta pemberitahuan untuk menghubungi ahli gizi untuk menu makanan yang lebih tepat | menampilkan menu makanan yang sesuai dengan kebutuhan kalori lebih dari atau sama dengan 2550 kalori bagi wanita beserta pemberitahuan untuk menghubungi ahli gizi untuk menu makanan yang lebih tepat | Valid |
| 8 | Memanggil fungsi lihat menu makanan bagi seorang wanita dengan kebutuhan kalori diantara | Berhasil menampilkan menu makanan sesuai dengan kebutuhan kalori diantara 2550 dan 2250 kalori bagi wanita | menampilkan menu makanan yang sesuai dengan kebutuhan kalori diantara 2550 dan 2250 kalori bagi wanita | Valid |

| | | | | |
|----|--|--|---|-------|
| | 2550 dan 2250 kalori | | | |
| 9 | Memanggil fungsi lihat menu makanan bagi seorang wanita dengan kebutuhan kalori kurang dari atau sama dengan 2250 kalori | Berhasil menampilkan menu makanan sesuai dengan kebutuhan kalori kurang dari atau sama dengan 2250 kalori bagi wanita | menampilkan menu makanan yang sesuai dengan kebutuhan kalori kurang dari atau sama dengan 2250 kalori bagi wanita | Valid |
| 10 | Memanggil fungsi lihat menu makanan bagi seorang wanita dengan kebutuhan kalori kurang dari 2250 kalori | Berhasil menampilkan menu makanan sesuai dengan kebutuhan kalori kurang dari 2250 kalori bagi pria beserta pemberitahuan untuk menghubungi ahli gizi untuk menu makanan yang lebih tepat | menampilkan menu makanan yang sesuai dengan kebutuhan kalori kurang dari atau sama dengan 2250 kalori bagi wanita beserta pemberitahuan untuk menghubungi ahli gizi untuk menu makanan yang lebih tepat | Valid |

6.1.2 Pengujian Unit Fitur Menghitung Kebutuhan Kalori Perhari

1. Pseudocode

Tabel 6.3 merupakan deskripsi dari *pseudocode* dari fitur menghitung kebutuhan kalori perhari

Tabel 6.3 Algoritma Fitur Menghitung Kebutuhan Kalori Perhari

| no | pseudocode |
|----|--|
| 1 | TextView = Text1.findViewById(R.id.pemberitahuan); ① |
| 2 | public void hitungkalori (View view){ |
| 3 | if (Cekissian() != false){ ② |
| 4 | nilaiberatbadan = |
| 5 | Integer.parseInt(beratbadan.getText().toString()); |
| 6 | nilaitinggibadan = |
| 7 | Integer.parseInt(tinggibadan.getText().toString()); |
| 8 | nilaiumur = Integer.parseInt(umur.getText().toString()); |
| 9 | //metodeharrisbenedict |
| 10 | double AMB = 655+ (9.6 * nilaiberatbadan) + (1.8 * |
| 11 | nilaitinggibadan) - (4.7 * nilaiumur); |
| 12 | double energi = AMB * nilaistress * nilaiaktivitas; ③ |

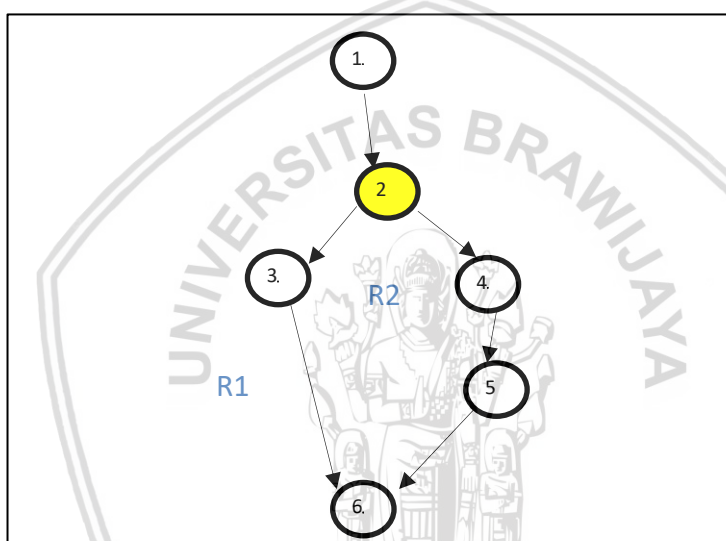
```

13     hasil = String.valueOf(energi);
14     hitung.setText(hasil);
15     int i= (int) Math.round(energi);
16     model_userhelper.tambahkaloriuser(sqliteDatabaseObj2
17     ,i,sessionemail);
18     Intent intent = new Intent(controller_hitungkalori.this,
19     controller_profil.class);
20     startActivity(intent);
21 }
22 Else { 4
23 Text1. setText("Harap Lengkapi data terlebih dahulu"); 5
24 }
25 End if 6
26 }

```

2. Basis Path Testing

a. Flow Graph



Gambar 6.2 Flow Graph Fitur Menghitung Kebutuhan Kalori Perhari

b. Cyclomatic Complexity

Dari gambar 6.2 diatas dapat kita tentukan berapa nilai $V(G)$ dengan ketentuan rumus sebagai berikut:

$$V(G) = \text{Edge} + \text{Nodes} + 2$$

$$V(G) = \text{Region}$$

$$V(G) = \text{Predicate note} + 1$$

Dari rumus diatas didapat lah hasil sebagai berikut berdasarkan gambar 6.2 *flow graph* menghitung ken yang telah dibuat

$$V(G) = 6 \text{ edges} - 6 \text{ nodes} + 2 = 2$$

$$V(G) = 2, \text{ ada } 2 \text{ region}$$

$$V(G) = 1 \text{ Predicate note} + 1 = 2$$

c. Independen *Path*

Jalur 1 = 1-2-3-6

Jalur 2 = 1-2-4-5-6

3. Hasil Pengujian Unit Fitur Menampilkan Menu Makanan

Tabel 6.4 Hasil Pengujian Unit Menghitung Kalori

| No Jalur | Prosedur Jalur | Expected Result | Result | Status |
|----------|--|--|--|--------|
| 1 | Melengkapi <i>form</i> hitung kalori lalu Memanggil fungsi menghitung kalori | Berhasil menampilkan hasil kebutuhan kalori | menampilkan hasil kebutuhan kalori | Valid |
| 2 | Tidak melengkapi <i>form</i> hitung kalori lalu Memanggil fungsi menghitung kalori | Menampilkan pemberitahuan untuk melengkapi <i>form</i> | menampilkan pemberitahuan untuk melengkapi <i>form</i> | Valid |

6.1.3 Pengujian Unit Fitur Melakukan *Login*

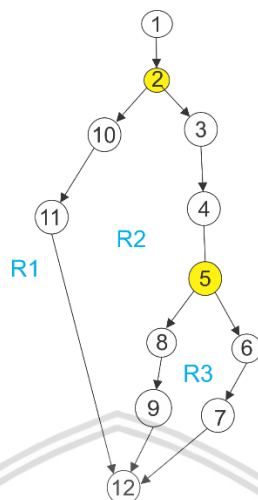
1. *Pseudocode*

Tabel 6.5 merupakan *pseudocode* dari fitur melakukan *login* yang dilakukan oleh pengguna

Tabel 6.5 Algoritma Fitur Melakukan *Login*

| no | <i>pseudocode</i> |
|----|---|
| 1 | String email = getinputanemail(); 1 |
| 2 | String pw = getinputanpassword(); |
| 3 | If(email != null password !=) 2 |
| 4 | Cek email pada database 3 |
| 5 | If (email terdaftar pada database){ 4 |
| 6 | cekpw(pw); 5 |
| 7 | If(cekpw==true){ 6 |
| 8 | Redirect home 7 |
| 9 | } |
| 10 | Else{ 8 |
| 11 | Toast.makeText("Tolong masukan email dan password yang |
| 12 | tepat.", Toast.LENGTH_LONG).show(); 9 |
| 13 | } |
| 14 | } |
| 15 | Else{ 10 |
| 16 | Toast.makeText("Tolong melengkapi email dan password pada form.", |
| 17 | Toast.LENGTH_LONG).show(); 11 |
| 18 | } |
| 19 | End if 12 |
| | } |

2. *Basis Path Testing*
a. *Flow Graph*



Gambar 6.3 Flow Graph Fitur Melakukan Login

b. *Cyclomatic Complexity*

Dari gambar 6.3 diatas dapat kita tentukan berapa nilai $V(G)$ dengan ketentuan rumus sebagai berikut:

$$V(G) = \text{Edge} + \text{Nodes} + 2$$

$$V(G) = \text{Region}$$

$$V(G) = \text{Predicate note} + 1$$

Dari rumus diatas didapat lah hasil sebagai berikut berdasarkan gambar 6.2 *flow graph* menghitung ken yang telah dibuat

$$V(G) = 13 \text{ edges} - 12 \text{ nodes} + 2 =$$

$$V(G) = 3, \text{ ada } 3 \text{ region}$$

$$V(G) = 2 \text{ Predicate note} + 1 = 4$$

c. *Independen Path*

Jalur 1 = 1-2-10-11-12

Jalur 2 = 1-2-3-4-5-8-9-12

Jalur 3 = 1-2-3-4-5-6-7-12

3. Hasil Pengujian Unit Fitur Menampilkan Menu Makanan

Tabel 6.6 Hasil Pengujian Unit Menampilkan Biodata User

| No Jalur | Prosedur Jalur | Expected Result | Result | Status |
|----------|--|---|---|--------|
| 1 | Mengisi inputan email dan password dengan tidak lengkap lalu memanggil fungsi <i>login</i> | Berhasil memberikan pemberitahuan inputan tidak lengkap | Berhasil memberikan pemberitahuan inputan tidak lengkap | Valid |

| | | | | |
|---|---|---|---|-------|
| 2 | Mengisi inputan email dan password dengan email dan password yang tidak terdaftar | Berhasil menampilkan pemberitahuan <i>email atau password salah</i> | Berhasil menampilkan pemberitahuan <i>email atau password salah</i> | Valid |
| 3 | Mengisi inputan email dan password dengan email dan password yang terdaftar | Berhasil melakukan <i>login</i> dan redirect ke halaman beranda | Berhasil melakukan <i>login</i> dan redirect ke halaman beranda | Valid |

6.2 Pengujian Validasi

Pengujian validasi merupakan salah satu dari metode pengujian *blackbox*, yaitu pengujian tanpa menghiraukan jalur algoritma yang berjalan pada suatu proses. Pengujian Validasi dilakukan dengan membuat kasus uji pada tiap fungsi fitur.

6.2.1 Kasus Uji Validasi

Berikut adalah skenario dari kasus uji aplikasi konsultasi menu makanan bagi penderita penyakit diabetes menggunakan metode *harris benedict*. Kasus Uji Melakukan Registrasi.

1. Kasus Uji Melakukan Registrasi

Pada tabel 6.7 dijelaskan lebih rinci bagaimana kasus uji dari fitur registrasi yang sudah diimplementasi.

Tabel 6.7 Tabel Kasus Uji Melakukan Registrasi

| | |
|------------------|---|
| Nama Kasus Uji | Kasus Uji Melakukan Registrasi |
| Objek Uji | Kebutuhan fungsional FR_1 |
| Tujuan Pengujian | Pengujian dilakukan untuk memastikan aplikasi dapat memenuhi kebutuhan fungsional melakukan registrasi pengguna |
| Prosedur Uji | <ol style="list-style-type: none"> 1. Pengguna memilih pilhan masuk sebagai <i>user</i> pada halaman <i>splashscreen</i> 2. Pengguna menekan tombol <i>register</i> pada halaman <i>login</i> 3. Pengguna memasukkan data sesuai dengan isian form yang sudah disiapkan 4. Pengguna menekan tombol simpan |

| | |
|-----------------------|---|
| Hasil Yang Diharapkan | Aplikasi berhasil melakukan registrasi pengguna lalu menyimpan data pengguna kedalam database |
|-----------------------|---|

2. Kasus Uji Melakukan *Login*

Pada tabel 6.8 dijelaskan lebih rinci bagaimana kasus uji dari fitur melakukan *login* yang sudah dimplementasi

Tabel 6.8 Tabel Kasus Uji Melakukan *Login*

| | |
|-----------------------|--|
| Nama Kasus Uji | Kasus Uji Melakukan <i>Login</i> |
| Objek Uji | Kebutuhan fungsional FR_2 |
| Tujuan Pengujian | Pengujian dilakukan untuk memastikan aplikasi dapat memenuhi kebutuhan fungsional dari fitur melakukan <i>login</i> pengguna |
| Prosedur Uji | <ol style="list-style-type: none"> 1. Pengguna memilih pilihan masuk sebagai <i>user</i> pada halaman <i>splashscreen</i> 2. Pengguna memasukan <i>email</i> dan <i>password</i> pada halaman <i>login</i> sesuai dengan proses registrasi yang sudah dilakukan sebelumnya 3. Pengguna menekan tombol masuk |
| Hasil Yang Diharapkan | Aplikasi berhasil mengarahkan pengguna ke halaman beranda |

3. Kasus Uji Menambahkan Biodata

Pada tabel 6.9 dijelaskan lebih rinci bagaimana kasus uji dari fitur menambahkan biodata yang sudah dimplementasi.

Tabel 6.9 Kasus Uji Menambahkan Biodata

| | |
|-----------------------|---|
| Nama Kasus Uji | Kasus Uji Menambahkan Biodata |
| Objek Uji | Kebutuhan fungsional FR_3 |
| Tujuan Pengujian | Pengujian dilakukan untuk memastikan aplikasi dapat memenuhi kebutuhan fungsional menambahkan biodata <i>pengguna</i> |
| Prosedur Uji | <ol style="list-style-type: none"> 1. Pengguna sudah berhasil registrasi 2. Pengguna memasukan biodata tinggibadan, berat badan, tanggalahir 3. Pengguna menekan tombol simpan |
| Hasil Yang Diharapkan | Aplikasi berhasil menambahkan biodata pengguna ke dalam database |

4. Kasus Uji Memperbarui Biodata

Pada tabel 6.10 dijelaskan lebih rinci bagaimana kasus uji dari fitur memperbarui biodata yang sudah dimplemetasi.

Tabel 6.10 Kasus Uji Memperbarui Biodata

| | |
|-----------------------|--|
| Nama Kasus Uji | Kasus Uji Menambahkan Biodata |
| Objek Uji | Kebutuhan fungsional FR_4 |
| Tujuan Pengujian | Pengujian dilakukan untuk memastikan aplikasi dapat memenuhi kebutuhan fungsional memperbarui biodata <i>pengguna</i> |
| Prosedur Uji | <ol style="list-style-type: none"> 1. Pengguna sudah berhasil <i>login</i> kedalam aplikasi sebagai <i>user</i> 2. Pengguna memilih pilihan profil 3. Pengguna memilih pilihan edit profil 4. Pengguna mengisi form biodata untuk di perbarui 5. Pengguna menekan tombol simpan |
| Hasil Yang Diharapkan | Aplikasi berhasil memperbarui biodata yang baru dari pengguna ke dalam database |

5. Kasus Uji Memperbarui Profil

Pada tabel 6.11 dijelaskan lebih rinci bagaimana kasus uji dari fitur memperbarui profil yang sudah dimplemetasi.

Tabel 6.11 Kasus Uji Menambahkan Profil

| | |
|-----------------------|--|
| Nama Kasus Uji | Kasus Uji Memperbarui Profil |
| Objek Uji | Kebutuhan fungsional FR_5 |
| Tujuan Pengujian | Pengujian dilakukan untuk memastikan aplikasi dapat memenuhi kebutuhan fungsional memperbarui biodata <i>pengguna</i> |
| Prosedur Uji | <ol style="list-style-type: none"> 1. Pengguna sudah berhasil <i>login</i> kedalam aplikasi sebagai <i>user</i> 2. Pengguna memilih pilihan profil 3. Pengguna memilih pilihan edit profil 4. Pengguna mengisi form edit profil untuk di perbarui 5. Pengguna menekan tombol simpan |
| Hasil Yang Diharapkan | Aplikasi berhasil memperbarui profil yang baru dari pengguna ke dalam database |

6. Kasus Uji Menghitung Kebutuhan Kalori Perhari

Pada tabel 6.12 dijelaskan lebih rinci bagaimana kasus uji dari fitur menghitung kebutuhan kalori perhari yang sudah diimplementasi.

Tabel 6.12 Kasus uji Uji Menghitung Kebutuhan Kalori Perhari

| | |
|-----------------------|--|
| Nama Kasus Uji | Kasus Uji Menghitung Kalori Perhari |
| Objek Uji | Kebutuhan fungsional FR_6 |
| Tujuan Pengujian | Pengujian dilakukan untuk memastikan aplikasi dapat memenuhi kebutuhan fungsional menghitung kalori perhari pengguna |
| Prosedur Uji | <ol style="list-style-type: none"> 1. Pengguna berhasil masuk sebagai <i>user</i> 2. Pengguna menekan tombol hitung kalori 3. Pengguna memasukkan biodata tinggi badan, berat badan dan umur 4. Pengguna memilih satu pilihan dari <i>form</i> pengukuran aktivitas 5. pengguna memilih satu pilihan dari <i>form</i> pengukuran stress 6. pengguna menekan tombol hitung kalori |
| Hasil Yang Diharapkan | Aplikasi berhasil menampilkan hasil perhitungan kalori yang dibutuhkan oleh pengguna |

7. Kasus Uji Melihat Saran Menu Makanan

Pada tabel 6.13 dijelaskan lebih rinci bagaimana kasus uji dari fitur melihat saran menu makanan yang sudah diimplementasi.

Tabel 6.13 Kasus uji Menampilkan Saran menu Makanan

| | |
|-----------------------|--|
| Nama Kasus Uji | Kasus Uji Menghitung Kalori Perhari |
| Objek Uji | Kebutuhan fungsional FR_7 |
| Tujuan Pengujian | Pengujian dilakukan untuk memastikan aplikasi dapat memenuhi kebutuhan fungsional menampilkan saran menu makanan |
| Prosedur Uji | <ol style="list-style-type: none"> 1. Pengguna berhasil masuk sebagai <i>user</i> 2. Pengguna telah melakukan perhitungan kalori 3. pengguna menekan tombol profil untuk melihat saran menu makanan |
| Hasil Yang Diharapkan | Aplikasi berhasil menampilkan saran menu makanan sesuai kalori kebutuhan pengguna |

8. Kasus Uji Menampilkan *List* Ahli Gizi

Pada tabel 6.14 dijelaskan lebih rinci bagaimana kasus uji dari fitur menampilkan *List* ahli gizi yang sudah dimplemetasi.

Tabel 6.14 Kasus uji Menampilkan *List* Ahli Gizi

| | |
|-----------------------|--|
| Nama Kasus Uji | Kasus Uji Melihat <i>List Ahli Gizi</i> |
| Objek Uji | Kebutuhan fungsional FR_8 |
| Tujuan Pengujian | Pengujian dilakukan untuk memastikan aplikasi dapat memenuhi kebutuhan fungsional menampilkan <i>List</i> ahli gizi |
| Prosedur Uji | <ol style="list-style-type: none"> 1. Pengguna berhasil <i>login</i> 2. Pengguna menekan tombol home |
| Hasil Yang Diharapkan | Aplikasi berhasil menampilkan <i>List</i> ahli gizi, sesuai dengan ahli gizi yang mendaftar pada aplikasi |

9. Kasus Uji Mengirim Pesan ke Ahli Gizi

Pada tabel 6.15 dijelaskan lebih rinci bagaimana kasus uji dari fitur mengirim pesan ke ahli gizi yang sudah dimplemetasi.

Tabel 6.15 Kasus Uji Mengirim Pesan ke Ahli Gizi

| | |
|-----------------------|---|
| Nama Kasus Uji | Kasus Uji Mengirim Pesan ke Ahli Gizi |
| Objek Uji | Kebutuhan fungsional FR_9 |
| Tujuan Pengujian | Pengujian dilakukan untuk memastikan aplikasi dapat memenuhi kebutuhan fungsional mengirim pesan keahligizi |
| Prosedur Uji | <ol style="list-style-type: none"> 1. Pengguna berhasil <i>login</i> 2. Pengguna memilih ahli gizi yang ingin dikirim pesan pada halaman beranda 3. pengguna menekan tombol tulis pesan 4. pengguna menuliskan konten pesan 5. pengguna menekan tombol kirim |
| Hasil Yang Diharapkan | Aplikasi berhasil mengirimkan pesan ke ahli gizi |

10. Kasus Uji Membalas Pesan ke Ahli Gizi

Pada tabel 6.16 dijelaskan lebih rinci bagaimana kasus uji dari fitur membalas pesan ke ahli gizi yang sudah dimplemetasi.

Tabel 6.16 Kasus Uji Membalas Pesan ke Ahli Gizi

| | |
|----------------|---------------------------------------|
| Nama Kasus Uji | Kasus Uji Membalas Pesan ke Ahli Gizi |
|----------------|---------------------------------------|

| | |
|-----------------------|---|
| Objek Uji | Kebutuhan fungsional FR_10 |
| Tujuan Pengujian | Pengujian dilakukan untuk memastikan aplikasi dapat memenuhi kebutuhan fungsional mengirim pesan ke ahli gizi |
| Prosedur Uji | <ol style="list-style-type: none"> 1. Pengguna berhasil <i>login</i> 2. Pengguna menekan tombol pesan 3. Pengguna memilih pesan yang ingin dibalas 4. pengguna memilih tombol tulis pesan 5. pengguna menuliskan alamat email ahli gizi 6. pengguna menuliskan konten pesan 7. pengguna menekan tombol kirim |
| Hasil Yang Diharapkan | Aplikasi berhasil mengirimkan pesan balasan ke ahli gizi |

11. Kasus Uji Melakukan *Logout*

Pada tabel 6.17 dijelaskan lebih rinci bagaimana kasus uji dari fitur melakukan *logout* yang sudah diimplementasi.

Tabel 6.17 Kasus Uji Melakukan *Logout*

| | |
|-----------------------|--|
| Nama Kasus Uji | Kasus Uji Melakukan <i>Logout</i> |
| Objek Uji | Kebutuhan fungsional FR_11 |
| Tujuan Pengujian | Pengujian dilakukan untuk memastikan aplikasi dapat memenuhi kebutuhan fungsional melakukan <i>logout</i> dari akun pengguna |
| Prosedur Uji | <ol style="list-style-type: none"> 1. Pengguna berhasil <i>login</i> 2. Pengguna menekan tombol <i>logout</i> pada halaman beranda |
| Hasil Yang Diharapkan | Aplikasi berhasil melakukan <i>logout</i> terhadap akun pengguna |

12. Kasus Uji Melakukan Registrasi Ahli Gizi

Pada tabel 6.18 dijelaskan lebih rinci bagaimana kasus uji dari fitur registrasi dari ahli gizi yang sudah diimplementasi.

Tabel 6.18 Tabel Kasus Uji Melakukan Registrasi Ahli Gizi

| | |
|------------------|--|
| Nama Kasus Uji | Kasus Uji Melakukan Registrasi Ahli Gizi |
| Objek Uji | Kebutuhan fungsional FR_12 |
| Tujuan Pengujian | Pengujian dilakukan untuk memastikan aplikasi dapat memenuhi kebutuhan fungsional melakukan registrasi ahli gizi |

| | |
|-----------------------|--|
| Prosedur Uji | <ol style="list-style-type: none"> 1. Ahli gizi memilih pilhan masuk sebagai Ahli gizi pada halaman <i>splashscreen</i> 2. Ahli gizi menekan tombol <i>register</i> pada halaman <i>login</i> 3. Ahli gizi memasukan data sesuai dengan isian form yang sudah disiapkan 4. Ahli gizi menekan tombol simpan |
| Hasil Yang Diharapkan | Aplikasi berhasil melakukan registrasi Ahli gizi lalu menyimpan data Ahli gizi kedalam database |

13. Kasus Uji Melakukan *Login* Ahli Gizi

Pada tabel 6.19 dijelaskan lebih rinci bagaimana kasus uji dari fitur melakukan *login* ahli gizi yang sudah dimplemetasi

Tabel 6.19 Tabel Kasus Uji Melakukan *Login* Ahli Gizi

| | |
|-----------------------|--|
| Nama Kasus Uji | Kasus Uji Melakukan <i>Login</i> Ahli Gizi |
| Objek Uji | Kebutuhan fungsional FR_13 |
| Tujuan Pengujian | Pengujian dilakukan untuk memastikan aplikasi dapat memenuhi kebutuhan fungsional dari fitur melakukan <i>login</i> pengguna |
| Prosedur Uji | <ol style="list-style-type: none"> 1. Ahli gizi memilih pilihan masuk sebagi Ahli gizi pada halaman <i>splashscreen</i> 2. Ahli gizi memasukan <i>email</i> dan <i>password</i> pada halaman <i>login</i> sesuai dengan proses registrasi yang sudah dilakukan sebelumnya 3. Ahli gizi menekan tombol masuk |
| Hasil Yang Diharapkan | Aplikasi berhasil mengarahkan Ahli gizi ke halaman beranda |

14. Kasus Uji Menambahkan Biodata Ahli Gizi

Pada tabel 6.20 dijelaskan lebih rinci bagaimana kasus uji dari fitur menambahkan biodata ahli gizi yang sudah dimplemetasi.

Tabel 6.20 Kasus Uji Menambahkan Biodata Ahli Gizi

| | |
|------------------|---|
| Nama Kasus Uji | Kasus Uji Menambahkan Biodata Ahli Gizi |
| Objek Uji | Kebutuhan fungsional FR_14 |
| Tujuan Pengujian | Pengujian dilakukan untuk memastikan aplikasi dapat memenuhi kebutuhan fungsional menambahkan biodata Ahli gizi |
| Prosedur Uji | <ol style="list-style-type: none"> 1. Ahli gizi sudah berhasil registrasi 2. Ahli gizi memasukan biodata tinggibadan, |

| | |
|-----------------------|---|
| | berat badan, tanggalahir 3. Ahli gizi menekan tombol simpan |
| Hasil Yang Diharapkan | Aplikasi berhasil menambahkan biodata Ahli gizi ke dalam database |

15. Kasus Uji Memperbarui Biodata Ahli gizi

Pada tabel 6.21 dijelaskan lebih rinci bagaimana kasus uji dari fitur memperbarui biodata Ahli gizi yang sudah dimplemetasi.

Tabel 6.21 Kasus Uji Memperbarui Biodata Ahli gizi

| | |
|-----------------------|---|
| Nama Kasus Uji | Kasus Uji Menambahkan Biodata Ahli gizi |
| Objek Uji | Kebutuhan fungsional FR_15 |
| Tujuan Pengujian | Pengujian dilakukan untuk memastikan aplikasi dapat memenuhi kebutuhan fungsional memperbarui biodata Ahli gizi |
| Prosedur Uji | <ol style="list-style-type: none"> 1. Ahli gizi sudah berhasil <i>login</i> kedalam aplikasi sebagai <i>user</i> 2. Ahli gizi memilih pilihan profil 3. Ahli gizi memilih pilihan edit profil 4. Ahli gizi mengisi form biodata untuk di perbarui 5. Ahli gizi menekan tombol simpan |
| Hasil Yang Diharapkan | Aplikasi berhasil memperbarui biodata yang baru dari Ahli gizi ke dalam database |

16. Kasus Uji Membalas Pesan ke Pengguna

Pada tabel 6.22 dijelaskan lebih rinci bagaimana kasus uji dari fitur membalas pesan ke pengguna yang sudah dimplemetasi.

Tabel 6.22 Kasus Uji Membalas Pesan ke Pengguna

| | |
|------------------|--|
| Nama Kasus Uji | Kasus Uji Membalas Pesan ke Ahli Gizi |
| Objek Uji | Kebutuhan fungsional FR_16 |
| Tujuan Pengujian | Pengujian dilakukan untuk memastikan aplikasi dapat memenuhi kebutuhan fungsional membalas pesan ke pengguna |
| Prosedur Uji | <ol style="list-style-type: none"> 1. Ahli gizi berhasil <i>login</i> sebagai ahli gizi 2. Ahli gizi memilih pesan yang ingin dibalas 3. Ahli gizi memilih tombol tulis pesan 4. Ahli gizi menuliskan alamat email pengguna 5. Ahli gizi menuliskan konten pesan 6. Ahli gizi menekan tombol kirim |

| | |
|-----------------------|--|
| Hasil Yang Diharapkan | Aplikasi berhasil mengirimkan pesan balasan ke ahli gizi |
|-----------------------|--|

17. Kasus Uji Menghitung Kebutuhan Kalori Perhari

Pada tabel 6.12 dijelaskan lebih rinci bagaimana kasus uji dari fitur menghitung kebutuhan kalori perhari yang dilakukan oleh ahli gizi yang sudah diimplementasi.

Tabel 6.23 Kasus uji Uji Menghitung Kebutuhan Kalori Perhari

| | |
|-----------------------|---|
| Nama Kasus Uji | Kasus Uji Menghitung Kalori Perhari |
| Objek Uji | Kebutuhan fungsional FR_6 |
| Tujuan Pengujian | Pengujian dilakukan untuk memastikan aplikasi dapat memenuhi kebutuhan fungsional menghitung kalori perhari pengguna |
| Prosedur Uji | <ol style="list-style-type: none"> 1. Pengguna berhasil masuk sebagai <i>user</i> 2. Pengguna menekan tombol hitung kalori 3. Pengguna memasukan biodata tinggibadan, berat badan dan umur 4. Pengguna memilih satu pilihan dari <i>form</i> pengukuran aktivitas 5. pengguna memilih satu pilihan dari <i>form</i> pengukuran stress 6. pengguna menekan tombol hitung kalori 7. aplikasi menampilkan hasil dari kebutuhan kalori perhari |
| Hasil Yang Diharapkan | Aplikasi berhasil menampilkan hasil perhitungan kebutuhan kalori perhari |

18. Kasus Uji Melakukan Logout Ahli Gizi

Pada tabel 6.17 dijelaskan lebih rinci bagaimana kasus uji dari fitur melakukan *logout* yang sudah diimplementasi.

Tabel 6.24 Kasus Uji Melakukan Logout Ahli Gizi

| | |
|------------------|--|
| Nama Kasus Uji | Kasus Uji Menghitung Kalori Perhari |
| Objek Uji | Kebutuhan fungsional FR_11 |
| Tujuan Pengujian | Pengujian dilakukan untuk memastikan aplikasi dapat memenuhi kebutuhan fungsional melakukan <i>logout</i> dari akun pengguna |
| Prosedur Uji | <ol style="list-style-type: none"> 1. Pengguna berhasil <i>login</i> 2. Pengguna menekan tombol <i>logout</i> pada halaman beranda |

| | |
|-----------------------|--|
| Hasil Yang Diharapkan | Aplikasi berhasil melakukan <i>logout</i> terhadap akun pengguna |
|-----------------------|--|

6.2.2 Analisa dan Hasil Pengujian Validasi

Tabel 6.25 merupakan hasil dari pengujian validasi berdasarkan kasus uji yang telah dilakukan sebelumnya.

Tabel 6.25 Hasil Pengujian Validasi

| Nomor Fr | Kasus Uji | Hasil yang Diharapkan | Hasil yang Didapat | Status |
|----------|--------------------------------|--|--|--------|
| FR_1 | Kasus Uji Melakukan Registrasi | Aplikasi berhasil melakukan registrasi pengguna lalu menyimpan data pengguna kedalam database | Aplikasi berhasil melakukan registrasi pengguna lalu menyimpan data pengguna kedalam database | Valid |
| FR_2 | Kasus Uji Melakukan Login | Aplikasi berhasil melakukan <i>login</i> dengan email dan password yang sesuai dengan proses registrasi pengguna | Aplikasi berhasil melakukan <i>login</i> dengan email dan password yang sesuai dengan proses registrasi pengguna | Valid |
| FR_3 | Kasus Uji Menambah kan Biodata | Aplikasi berhasil menambahkan biodata pengguna ke dalam database | Aplikasi berhasil menambahkan biodata <i>user</i> ke dalam database | Valid |
| FR_4 | Kasus Uji Memperbar ui Biodata | Aplikasi berhasil memperbarui biodata pengguna ke dalam database | Aplikasi berhasil memperbarui biodata pengguna ke dalam database | Valid |
| FR_5 | Kasus Uji Memperbar ui profil | Aplikasi berhasil memperbarui profil pengguna ke dalam database | Aplikasi berhasil memperbarui profil pengguna ke dalam database | Valid |
| FR_6 | Kasus Uji Menghitung kebutuhan | Aplikasi berhasil menampilkan hasil perhitungan kalori | Aplikasi berhasil menampilkan hasil perhitungan kalori yang | Valid |

| | | | | |
|-------|--|--|--|-------|
| | Kalori perhari | yang dibutuhkan oleh pengguna | dibutuhkan oleh pengguna | |
| FR_7 | Kasus Uji Melihat Saran Menu Makanan | Aplikasi berhasil melihat saran menu makanan sesuai kalori kebutuhan pengguna | Aplikasi berhasil melihat saran menu makanan sesuai kalori kebutuhan pengguna | Valid |
| FR_8 | Kasus Uji Melihat <i>List Ahli Gizi</i> | Aplikasi berhasil melihat <i>List ahli gizi</i> , sesuai dengan ahli gizi yang mendaftar pada aplikasi | Aplikasi berhasil melihat <i>List ahli gizi</i> , sesuai dengan ahli gizi yang mendaftar pada aplikasi | Valid |
| FR_9 | Kasus Uji Mengirim Pesan ke Ahli Gizi | Aplikasi berhasil mengirimkan pesan ke ahli gizi | Aplikasi berhasil mengirimkan pesan ke ahli gizi | Valid |
| FR_10 | Kasus Uji Membalas Pesan ke Ahli Gizi | Aplikasi berhasil mengirimkan pesan balasan ke ahli gizi sesuai dengan email pengirim pesan | Aplikasi berhasil mengirimkan pesan balasan ke ahli gizi sesuai dengan email pengirim pesan | Valid |
| FR_11 | Kasus Uji Melakukan <i>Logout</i> | Aplikasi berhasil melakukan <i>logout</i> akun pengguna terhadap aplikasi dan diarahkan keluar ke menu <i>splashscreen</i> | Aplikasi berhasil melakukan <i>logout</i> akun pengguna terhadap aplikasi dan diarahkan keluar ke menu <i>splashscreen</i> | Valid |
| FR_12 | Kasus Uji Melakukan Registrasi Ahli Gizi | Aplikasi berhasil melakukan registrasi ahli gizi lalu menyimpan data pengguna kedalam database | Aplikasi berhasil melakukan registrasi ahli gizi lalu menyimpan data <i>user</i> kedalam database | Valid |
| FR_13 | Kasus Uji Melakukan <i>Login</i> Ahli Gizi | Aplikasi berhasil melakukan <i>login</i> dengan email dan password yang sesuai dengan | Aplikasi berhasil melakukan <i>login</i> dengan email dan password yang sesuai dengan | Valid |

| | | | | |
|-------|---|--|--|-------|
| | | proses registrasi Ahli Gizi | proses registrasi Ahli Gizi | |
| FR_14 | Kasus Uji Menambahkan Biodata Ahli Gizi | Aplikasi berhasil menambahkan biodata Ahli Gizi ke dalam database | Aplikasi berhasil menambahkan biodata Ahli Gizi ke dalam database | Valid |
| FR_15 | Kasus Uji Memperbarui Biodata Ahli Gizi | Aplikasi berhasil memperbarui biodata Ahli Gizi ke dalam database | Aplikasi berhasil memperbarui biodata Ahli Gizi ke dalam database | Valid |
| FR_16 | Kasus Uji Membalas Pesan ke pengguna | Aplikasi berhasil mengirimkan pesan balasan ke pengguna sesuai dengan email pengirim pesan | Aplikasi berhasil mengirimkan pesan balasan ke pengguna sesuai dengan email pengirim pesan | Valid |
| FR_17 | Kasus Uji Menghitung kebutuhan Kalori perhari Ahli Gizi | Aplikasi berhasil menampilkan hasil perhitungan kalori yang di lakukan oleh ahli gizi | Aplikasi berhasil menampilkan hasil perhitungan kalori yang di lakukan oleh ahli gizi | Valid |
| FR_18 | Kasus Uji Melakukan Logout | Aplikasi berhasil melakukan <i>logout</i> akun pengguna terhadap aplikasi dan diarahkan keluar ke menu <i>splashscreen</i> | Aplikasi berhasil melakukan <i>logout</i> akun pengguna terhadap aplikasi dan diarahkan keluar ke menu <i>splashscreen</i> | Valid |

6.3 Pengujian *Usability*

Pengujian *usability* aplikasi konsultasi menu makanan bagi penderita diabetes ini dilakukan pengambilan 10 sample pengguna aplikasi yang dibagi menjadi dua kategori *user* dan ahli gizi. Pengambilan 10 sample pengguna didasarkan kepada Gay dan Diehl, Roscoe (1975) yang memberikan panduan Untuk penelitian eksperimental sederhana ukuran sampel kecil antara 10 sampai dengan 20. Pengujian *usability* dilakukan untuk menguji kebutuhan nonfungsional aplikasi yaitu mudahnya aplikasi dipahami oleh pengguna sudah terpenuhi atau belum.

6.3.1 Kasus Uji *Usability*

User dan ahli gizi melakukan registrasi ke aplikasi dan mencoba fitur-fitur yang tersedia. Setelah itu *user* dan ahli gizi memberi respon terhadap 10 item pernyataan berdasarkan metode uji pengguna *System Usability Scale* (SUS).

Metode uji pengguna *System Usability Scale* ini menggunakan 10 item pernyataan yang dijelaskan pada tabel 6.26

Tabel 6.26 Tabel Kuisiner *System Usability Scale*

| No | Pertanyaan | Sangat Tidak Setuju | | | | Sangat Setuju |
|----|---|---------------------|---|---|---|---------------|
| | | 1 | 2 | 3 | 4 | 5 |
| 1 | Saya rasa saya akan sering menggunakan sistem ini. | | | | | |
| 2 | Saya merasa sistem terlalu kompleks padahal sebenarnya dapat dibuat sederhana. | | | | | |
| 3 | Saya rasa sistem mudah untuk digunakan. | | | | | |
| 4 | Saya rasa saya membutuhkan bantuan dari orang teknis untuk dapat menggunakan sistem ini. | | | | | |
| 5 | Saya menemukan bahwa terdapat berbagai macam fungsi yang terintegrasi dengan baik dalam sistem. | | | | | |
| 6 | Saya rasa banyak hal yang tidak konsisten terdapat pada sistem. | | | | | |
| 7 | Saya rasa mayoritas pengguna akan belajar menggunakan sistem ini secara cepat. | | | | | |
| 8 | Saya menemukan bahwa sistem sangat tidak praktis. | | | | | |
| 9 | Saya sangat percaya dalam menggunakan sistem ini. | | | | | |

| | | | | | | |
|----|--|--|--|--|--|--|
| 10 | Saya harus belajar banyak hal terlebih dahulu sebelum saya dapat menggunakan sistem ini. | | | | | |
|----|--|--|--|--|--|--|

Perhitungan bobot untuk masing masing item mempunyai aturan sebagai berikut:

1. Untuk item no 1,3,5,7, dan 9 nilai yang didapat adalah posisi skala dikurangi 1.
2. Untuk item 2,4,6,8 dan 10, nilai yang didapat adalah 5 dikurangi posisi skala
3. Kemudian jumlahkan nilai yang didapat dan lakukan proses perhitungan nilai jumlah dikalikan 2,5 untuk mendapatkan nilai keseluruhan skor SUS.

6.3.2 Analisa dan Hasil Pengujian *Usability* Pengguna

Hasil kuisioner yang telah diberikan kepada responden dijelaskan pada tabel 6.27 berikut.

Tabel 6.27 Hasil Kuisioner Pertanyaan SUS

| Nama | User 1 | User 2 | User 3 | User 4 | User 5 | User 6 | User 7 | User 8 | User 9 | User 10 |
|------|--------|--------|--------|--------|--------|--------|--------|--------|--------|---------|
| No | | | | | | | | | | |
| 1 | 4 | 3 | 4 | 3 | 4 | 2 | 4 | 3 | 4 | 4 |
| 2 | 3 | 2 | 2 | 3 | 3 | 2 | 4 | 2 | 3 | 2 |
| 3 | 2 | 3 | 3 | 4 | 3 | 3 | 4 | 4 | 3 | 4 |
| 4 | 3 | 3 | 3 | 2 | 4 | 3 | 4 | 3 | 3 | 3 |
| 5 | 4 | 3 | 2 | 4 | 4 | 3 | 4 | 4 | 3 | 4 |
| 6 | 3 | 2 | 3 | 2 | 3 | 4 | 1 | 4 | 2 | 3 |
| 7 | 2 | 4 | 1 | 3 | 4 | 4 | 3 | 4 | 3 | 1 |
| 8 | 3 | 3 | 2 | 4 | 1 | 4 | 3 | 1 | 2 | 3 |
| 9 | 2 | 3 | 2 | 4 | 3 | 2 | 3 | 4 | 4 | 2 |
| 10 | 3 | 2 | 4 | 3 | 4 | 4 | 3 | 1 | 4 | 3 |

Kemudian dari hasil kuisioner tersebut dilakukan konversi yang dijelaskan pada table 6.28 berikut.

Tabel 6.28 Hasil Konversi Kuisiener SUS

| Nama | User 1 | User 2 | User 3 | User 4 | User 5 | User 6 | User 7 | User 8 | User 9 | User 10 |
|----------|--------|--------|--------|--------|--------|--------|--------|--------|--------|---------|
| No | | | | | | | | | | |
| 1 | 10 | 7.5 | 10 | 7.5 | 10 | 5 | 10 | 7.5 | 10 | 10 |
| 2 | 7.5 | 5 | 5 | 7.5 | 7.5 | 5 | 10 | 5 | 7.5 | 5 |
| 3 | 5 | 7.5 | 7.5 | 10 | 7.5 | 7.5 | 10 | 10 | 7.5 | 10 |
| 4 | 7.5 | 7.5 | 7.5 | 5 | 10 | 7.5 | 10 | 7.5 | 7.5 | 7.5 |
| 5 | 10 | 7.5 | 5 | 10 | 10 | 7.5 | 10 | 10 | 7.5 | 10 |
| 6 | 7.5 | 5 | 7.5 | 5 | 7.5 | 10 | 2.5 | 10 | 5 | 7.5 |
| 7 | 5 | 10 | 2.5 | 7.5 | 10 | 10 | 7.5 | 10 | 7.5 | 2.5 |
| 8 | 7.5 | 7.5 | 5 | 10 | 2.5 | 10 | 7.5 | 2.5 | 5 | 7.5 |
| 9 | 5 | 7.5 | 5 | 10 | 7.5 | 5 | 7.5 | 10 | 10 | 5 |
| 10 | 7.5 | 5 | 10 | 7.5 | 10 | 10 | 7.5 | 2.5 | 10 | 7.5 |
| Rat Rata | 76.25 | | | | | | | | | |

Berdasarkan hasil dari kuisiener SUS dengan skor rata – rata 7.62 dari 10 orang responden dapat dikatakan aplikasi sudah masuk dalam kategori *acceptable*.

6.3.3 Analisa dan Hasil Pengujian *Usability* Ahli Gizi

Hasil kuisiener yang telah diberikan kepada responden dijelaskan pada table 6.29 berikut.

Tabel 6.29 Hasil Kuisiener Pertanyaan SUS Ahli Gizi

| Nama | Ahli Gizi 1 | Ahli Gizi 2 | Ahli Gizi 3 | Ahli Gizi 4 | Ahli Gizi 5 | Ahli Gizi 6 | Ahli Gizi 7 | Ahli Gizi 8 | Ahli Gizi 9 | Ahli Gizi 10 |
|------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|--------------|
| No | | | | | | | | | | |
| 1 | 3 | 4 | 2 | 3 | 4 | 4 | 4 | 4 | 4 | 3 |
| 2 | 1 | 1 | 1 | 3 | 4 | 3 | 3 | 1 | 3 | 4 |
| 3 | 4 | 2 | 3 | 4 | 3 | 3 | 3 | 3 | 3 | 3 |
| 4 | 3 | 3 | 3 | 1 | 3 | 3 | 4 | 3 | 3 | 3 |
| 5 | 4 | 4 | 3 | 4 | 2 | 3 | 4 | 2 | 4 | 3 |
| 6 | 4 | 3 | 4 | 2 | 2 | 2 | 3 | 3 | 3 | 2 |
| 7 | 4 | 4 | 4 | 3 | 3 | 3 | 4 | 4 | 3 | 4 |
| 8 | 1 | 3 | 4 | 4 | 3 | 2 | 1 | 2 | 3 | 3 |

| | | | | | | | | | | |
|----|---|---|---|---|---|---|---|---|---|---|
| 9 | 2 | 4 | 4 | 4 | 3 | 4 | 3 | 2 | 4 | 3 |
| 10 | 3 | 3 | 3 | 3 | 4 | 4 | 4 | 4 | 3 | 4 |

Kemudian dari hasil kuisioner tersebut dilakukan konversi yang dijelaskan pada table 6.30 berikut.

Tabel 6.30 Hasil Konversi Kuisioner SUS Ahli Gizi

| Nama No | Ahli Gizi 1 | Ahli Gizi 2 | Ahli Gizi 3 | Ahli Gizi 4 | Ahli Gizi 5 | Ahli Gizi 6 | Ahli Gizi 7 | Ahli Gizi 8 | Ahli Gizi 9 | Ahli Gizi 10 |
|------------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|--------------------|
| 1 | 7.5 | 10 | 5 | 7.5 | 10 | 10 | 10 | 10 | 10 | 7.5 |
| 2 | 2.5 | 2.5 | 2.5 | 7.5 | 10 | 7.5 | 7.5 | 2.5 | 7.5 | 10 |
| 3 | 10 | 5 | 7.5 | 10 | 7.5 | 7.5 | 7.5 | 7.5 | 7.5 | 7.5 |
| 4 | 7.5 | 7.5 | 7.5 | 2.5 | 7.5 | 7.5 | 10 | 7.5 | 7.5 | 7.5 |
| 5 | 10 | 10 | 7.5 | 10 | 5 | 7.5 | 10 | 5 | 10 | 7.5 |
| 6 | 10 | 7.5 | 10 | 5 | 5 | 5 | 7.5 | 7.5 | 7.5 | 5 |
| 7 | 10 | 10 | 10 | 7.5 | 7.5 | 7.5 | 10 | 10 | 7.5 | 10 |
| 8 | 2.5 | 7.5 | 10 | 10 | 7.5 | 5 | 2.5 | 5 | 7.5 | 7.5 |
| 9 | 5 | 10 | 10 | 10 | 7.5 | 10 | 7.5 | 5 | 10 | 7.5 |
| 10 | 7.5 | 7.5 | 7.5 | 7.5 | 10 | 10 | 10 | 10 | 7.5 | 10 |
| Total | 7.5 | 10 | 5 | 7.5 | 10 | 10 | 10 | 10 | 10 | 7.5 |
| Rat Rata | 7.5 | | | | | | | | | |

Berdasarkan hasil dari kuisioner SUS dengan skor rata – rata 7.5 dari 10 orang responden dapat dikatakan aplikasi sudah masuk dalam kategori *acceptable*.

BAB 7 PENUTUP

7.1 Kesimpulan

Dari hasil analisis kebutuhan, perancangan, implementasi, dan pengujian yang telah dilakukan sebelumnya maka dapat ditarik kesimpulan sebagai berikut:

1. Berdasarkan hasil analisis kebutuhan yang dilakukan, aplikasi konsultasi menu makanan menggunakan metode *harris benedict* ini memiliki 18 kebutuhan fungsional dan 1 kebutuhan non fungsional. Dimana kebutuhan – kebutuhan tersebut didapat dari permasalahan susahny penderita penyakit diabetes untuk menemukan konsultan menu makanan dan sulitnya lulusan – lulusan sarjana baru universitas brawijaya untuk menemukan pasien.
2. Kemudian setelah analisis kebutuhan dilakukan, dilanjutkan dengan perancangan arsitektural, perancangan perhitungan kalori menggunakan metode *harris benedict*, perancangan *sequence diagram* perancangan *class diagram* dan perancangan antar muka aplikasi.
3. Pemanfaatan metode *Harris benedict* dapat diandalkan pada saat aplikasi melakukan fitur menghitung kebutuhan kalori *user* berdasarkan *form* biodata yang telah disediakan oleh aplikasi.
4. Dari hasil pengujian *Usability* yang telah dilakukan didapatkan hasil pengujian dari sisi pengguna mendapatkan poin 7.62 dan dari sisi ahli gizi mendapatkan poin 7.5. Oleh karena itu dapat disimpulkan bahwa aplikasi ini dapat dikatakan memberikan kemudahan bagi pengguna.
5. Dari hasil penelitian yang dilakukan, pembuatan prototipe aplikasi konsultasi menu makanan menggunakan metode *harris benedict* dapat dikatakan membantu pasien penderita penyakit diabetes dan lulusan – lulusan baru ahli gizi ub.

7.2 Saran

Saran yang diberikan yang akan berguna dalam pengembangan aplikasi penentuan menu makanan menggunakan metode *harris benedict* ini antara lain:

1. Dapat menampilkan menu makanan yang lebih bervariasi.
2. Aplikasi dalam pengembangannya disarankan dapat menampilkan saran menu makanan berdasarkan kebutuhan yang lebih spesifik seperti vitamin, lemak, nutrisi dan lainnya.
3. Aplikasi dapat menyediakan fitur bagi *user* untuk menghitung kalori makanan yang telah *user* santap.

DAFTAR PUSTAKA

- Adam J. *Komplikasi Kronik Diabetik Masalah Utama Penderita Diabetes dan Upaya Pencegahan*. Tersedia di: <<http://www.akademik.unsri.ac.id/download/journal/files/medhas/9John%20Adam>> [Diakses 2 April 2017]
- Almatsier., 2008. *Prinsip Dasar Ilmu Gizi*. Jakarta: Gramedia Pustaka Umum.
- Bangor, A., Kortum, P., & Miller, J., 2009. Determining what individual SUS scores mean: Adding an adjective rating scale, [ejournal] 4(3). Tersedia melalui: *Journal of Usability Studies (JUS)*.
- Brooke, J., 1986, SUS-A Quick dan Dirty *Usability Scale*, *Usability Evaluation in Industry*, vol. 189.
- Depkes., 2012. *Diabetes di Indonesia*. [online] Tersedia di: <http://www.depkes.go.id/article/view/1314/diabetes-melitus-dapat-dicegah.html> [Diakses 4 April 2017]
- Dennis et al. 2012. *Systems Analysis & Design With UML Version 2.0; An Object-Oriented Approach 4th Edition*. John Wiley & Sons, Inc. Edition, McGraw-Hill Book, Co.
- Foster DW. *Diabetes Mellitus in Harrison Prinsip-Prinsip Ilmu Penyakit Dalam*. 13 ed. Jakarta: EGC; 1996.
- Herdianingsih Dida. *Pengertian, Penyebab, Gejala Dan Cara Pencegahan Penyakit Diabetes Melitus*. [online] Tersedia di: <http://www.sehatdengaherbal.com/pengertian-penyebab-gejala-dan-cara-pencegahan-penyakit-diabetes-melitus> [Diakses 4 April 2017]
- Herdianingsih, S., Muhtadi, A., Lestari, K., dan Annisa, N. 2016, Potensi Interaksi Obat-Obat pada Resep Polifarmasi: Studi Retrospektif pada Salah Satu Apotek di Kota Bandung, *Jurnal Farmasi Klinik Indonesia*.
- Irianto, Djoko Pekik 2007. *Panduan Gizi Lengkap Keluarga dan Olahragawan*. Andi Offset. Yogyakarta.
- Martono H PK, Rahayu RA, Joni B, Huda IS, Murti Y, 2007, Diabetes melitus pada lanjut usia. In: Darmono ST, dkk editor. *Naskah lengkap diabetes melitus*. Semarang: Badan Penerbit Universitas Diponegoro.
- Merk. *Kidney Disease of Diabetes*. National Institute of Diabetes and Digestive and Kidney Diseases; 20008; Tersedia di: <http://www.kidney.niddk.nih.gov/kudiseases/pubs/kdd/index.htm> [Diakses 2 April 2017]
- Nugroho.Adi, 2009. *Rekayasa Perangkat Lunak Menggunakan UML & Java*. Yogyakarta: Andi Offset.

Rochman, W, 2006. Diabetes Mellitus Pada Usia Lanjut, Jilid III, Edisi IV; Sudoyono A, W., Setiyohadi B., Alwi I., Simadibrata K.M., Setiati S. (eds), *Buku Ajar Ilmu Penyakit Dalam*, Balai Penerbit Fakultas Kedokteran Universitas Indonesia, Jakarta.

Soegondo, S., dan Soewondo, P., dan Subekti, I, 2005. *Penata Laksanaan Diabetes Mellitus Terpadu, Sebagai Panduan Penatalaksanaan Diabetes Mellitus bagi Dokter Maupun Edukator*, Balai Penerbit Fakultas Kedokteran Universitas Indonesia, Jakarta.

